

# Scalarized Preferences in Multi-objective Optimization

Zur Erlangung des akademischen Grades eines  
Doktors der Wirtschaftswissenschaften  
(Dr. rer. pol.)

bei der KIT-Fakultät für Wirtschaftswissenschaften  
des Karlsruher Instituts für Technologie (KIT)

genehmigte  
DISSERTATION

von

M.Sc. Marlon Alexander Braun

Datum der mündlichen Prüfung: 19.03.2018  
Referent: PD Dr. Pradyumn Shukla  
Korreferent: Prof. Dr. Oliver Stein



This document is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0):  
<https://creativecommons.org/licenses/by/4.0/deed.en>

# Abstract

Multi-objective optimization problems are characterized by possessing no single solution that attains the best feasible value in each objective. The intricacy of these problems lies in finding a compromise solution such that the preferences of the decision maker, who implements the compromise, are satisfied. Scalarization – mapping the vector of objectives to a real value – identifies a single solution as global preference optimum to solve this predicament, however, scalarization generates no information about other compromise solutions that might change the decision maker’s preferences towards the global optimum. To address this issue, this thesis provides a theoretical and algorithmic analysis of scalarized preferences. The theoretical analysis consists of the development of a framework that characterizes preferences as problem transformations that identify preferred subsets of the Pareto front. Scalarization is placed within this framework as a codomain transformation. Furthermore, axioms are proposed that represent desirable properties scalarization functions may exhibit. It is shown under which conditions existing scalarization functions fulfill these axioms. The algorithmic analysis characterizes preferences by the result an optimization algorithm generates. Two new paradigms are identified within the analysis for which algorithms that use scalarized preference information are designed: Preferences-biased Pareto front approximations distribute points across the entire Pareto front but focus more points in regions with better scalarization values; multimodal preference optima are points that are local scalarization optima in the objective space. A three step algorithm is developed to approximate local scalarization optima and different methods are evaluated for each step. Two real-world optimization problems are presented to illustrate the usefulness of the two algorithms. The first problem consists of finding operating schedules for a combined heat and power plant that maximize electricity and heat output while minimizing fuel consumption. Preference-biased approximations generate more energy efficient solutions among which the decision maker can choose her personal preference weighing the tradeoffs between the three objectives. The second problem is concerned with scheduling devices in a residential building such that energy costs, carbon dioxide emissions and thermal discomfort are minimized. It is shown that local scalarization optima are schedules that balance the three objectives. The analysis and the experiments presented in this work enable decision makers to make better choices by applying methods generating more options that conform to their preferences.



# Zusammenfassung

Multikriterielle Optimierungsprobleme verfügen über keine Lösung, die optimal in jeder Zielfunktion ist. Die Schwierigkeit solcher Probleme liegt darin eine Kompromisslösung zu finden, die den Präferenzen des Entscheiders genügen, der den Kompromiss implementiert. Skalarisierung – die Abbildung des Vektors der Zielfunktionswerte auf eine reelle Zahl – identifiziert eine einzige Lösung als globales Präferenzenoptimum um diese Probleme zu lösen. Allerdings generieren Skalarisierungsmethoden keine zusätzlichen Informationen über andere Kompromisslösungen, die die Präferenzen des Entscheiders bezüglich des globalen Optimums verändern könnten. Um dieses Problem anzugehen stellt diese Dissertation eine theoretische und algorithmische Analyse skalarisierter Präferenzen bereit. Die theoretische Analyse besteht aus der Entwicklung eines Ordnungsrahmens, der Präferenzen als Problemtransformationen charakterisiert, die präferierte Untermengen der Paretofront definieren. Skalarisierung wird als Transformation der Zielmenge in diesem Ordnungsrahmen dargestellt. Des Weiteren werden Axiome vorgeschlagen, die wünschenswerte Eigenschaften von Skalarisierungsfunktionen darstellen. Es wird gezeigt unter welchen Bedingungen existierende Skalarisierungsfunktionen diese Axiome erfüllen. Die algorithmische Analyse kennzeichnet Präferenzen anhand des Resultats, das ein Optimierungsalgorithmus generiert. Zwei neue Paradigmen werden innerhalb dieser Analyse identifiziert. Für beide Paradigmen werden Algorithmen entworfen, die skalarisierte Präferenzeninformationen verwenden: Präferenzen-verzerrte Paretofrontapproximationen verteilen Punkte über die gesamte Paretofront, fokussieren aber mehr Punkte in Regionen mit besseren Skalarisierungswerten; multimodale Präferenzenoptima sind Punkte, die lokale Skalarisierungsoptima im Zielraum darstellen. Ein Drei-Stufen-Algorithmus wird entwickelt, der lokale Skalarisierungsoptima approximiert und verschiedene Methoden werden für die unterschiedlichen Stufen evaluiert. Zwei Realweltprobleme werden vorgestellt, die die Nützlichkeit der beiden Algorithmen illustrieren. Das erste Problem besteht darin Fahrpläne für ein Blockheizkraftwerk zu finden, die die erzeugte Elektrizität und Wärme maximieren und den Kraftstoffverbrauch minimiert. Präferenzen-verzerrte Approximationen generieren mehr Energie-effiziente Lösungen, unter denen der Entscheider seine favorisierte Lösung auswählen kann, indem er die Konflikte zwischen den drei Zielen abwägt. Das zweite Problem beschäftigt sich mit der Erstellung von Fahrplänen für Geräte in einem Wohngebäude, so dass Energiekos-

ten, Kohlenstoffdioxidemissionen und thermisches Unbehagen minimiert werden. Es wird gezeigt, dass lokale Skalarisierungsoptima Fahrpläne darstellen, die eine gute Balance zwischen den drei Zielen bieten. Die Analyse und die Experimente, die in dieser Arbeit vorgestellt werden, ermöglichen es Entscheidern bessere Entscheidungen zu treffen indem Methoden angewendet werden, die mehr Optionen generieren, die mit den Präferenzen der Entscheider übereinstimmen.

*to Stefan, Simone and Nicola*



# Contents

<b>List of Figures</b>	<b>xiii</b>
<b>List of Tables</b>	<b>xvii</b>
<b>Glossary</b>	<b>xxi</b>
<b>Acronyms</b>	<b>xxv</b>
<b>Notation</b>	<b>xxix</b>
<b>1. Introduction</b>	<b>1</b>
1.1. Motivation . . . . .	1
1.2. Related Work . . . . .	4
1.3. Objectives and Contributions . . . . .	6
1.4. Structure . . . . .	7
<b>2. Preferences in Multi-objective Optimization</b>	<b>9</b>
2.1. Multi-objective Optimization Problems . . . . .	9
2.2. Preference Modeling . . . . .	16
2.3. Problem Transformations Using Tradeoff-based Preference Models	21
2.3.1. Tradeoff Domain Predicate . . . . .	22
2.3.2. Tradeoff Order Predicate . . . . .	24
2.3.3. Tradeoff Codomain Predicate . . . . .	26
<b>3. Scalarized Preferences</b>	<b>39</b>
3.1. The Scalarized Preference Model . . . . .	39
3.1.1. Additive Scalarization Function . . . . .	41
3.1.2. Multiplicative Scalarization Function . . . . .	42
3.1.3. Maximal Scalarization Function . . . . .	43
3.1.4. Weighted Sum and Sum of Objectives . . . . .	44
3.1.5. Weighted Product and Product of Objectives . . . . .	45
3.1.6. Chebyshev Method . . . . .	46
3.1.7. Nash Bargaining Solution . . . . .	48
3.1.8. Angle Utility . . . . .	49
3.1.9. Tradeoff Utility . . . . .	51

Contents

- 3.2. Characterizing Scalarization Functions . . . . . 52
  - 3.2.1. Pareto Compliance . . . . . 55
  - 3.2.2. Binary Independence . . . . . 58
  - 3.2.3. Non-Extremeness . . . . . 59
  - 3.2.4. Contraction Consistency . . . . . 65
  - 3.2.5. Monotonicity . . . . . 68
  - 3.2.6. Equity . . . . . 70
  - 3.2.7. Invariance to Scaling . . . . . 75
  
- 4. Methodologies for Solving Multi-objective Optimization Problems 83**
  - 4.1. Solving Paradigms in Multi-objective Optimization . . . . . 83
    - 4.1.1. Closed-form Expression . . . . . 86
    - 4.1.2. Finite Set of Points . . . . . 87
    - 4.1.3. Subset Approximation . . . . . 89
    - 4.1.4. Global Preference Optima . . . . . 90
    - 4.1.5. A Hierarchy of Solving Paradigms . . . . . 91
    - 4.1.6. Preference-biased Pareto front approximations . . . . . 94
    - 4.1.7. Multimodal preference optima . . . . . 95
    - 4.1.8. An Amended Hierarchy of Solving Paradigms . . . . . 95
  - 4.2. Nature-inspired Problem Solving . . . . . 97
  
- 5. Preference-biased Pareto Front Approximations 101**
  - 5.1. Theoretical Foundation of Preference-biased Approximations . . . 102
  - 5.2. Algorithmic Approach to Preference-biased Approximations . . . 105
  - 5.3. Computational Analysis of Preference-biased Approximations . . . 112
    - 5.3.1. Quantitative Comparison of Replacement Strategies . . . . . 113
    - 5.3.2. Quantitative Analysis of the No Preference Case . . . . . 144
    - 5.3.3. Qualitative Analysis of Preference-biased Approximations 183
  
- 6. Multimodal Scalarized Preferences 191**
  - 6.1. Local Scalarization Optima and Basins . . . . . 191
    - 6.1.1. Local Scalarization Optima . . . . . 192
    - 6.1.2. Basins of Attraction . . . . . 193
    - 6.1.3. Local Scalarization Optima Preference Predicate . . . . . 194
  - 6.2. Algorithmic Framework for Approximating Scalarization Optima 194
  - 6.3. Computational Analysis of Framework Components . . . . . 197
    - 6.3.1. Basin Identification Methods . . . . . 198
    - 6.3.2. Local Optimization Algorithms . . . . . 215
  
- 7. Real-world Applications 233**
  - 7.1. Combined Heat and Power Plant . . . . . 233
    - 7.1.1. Modeling of a Complex Cogeneration Process . . . . . 234
    - 7.1.2. Optimization of a Complex Cogeneration Process . . . . . 240

7.2. Building Energy Management Systems . . . . .	247
7.2.1. A Multi-objective Framework for Building Energy Man- agement Systems . . . . .	248
7.2.2. Optimal Operating Schedules in Residential Buildings . . . . .	259
<b>8. Conclusion and Outlook</b>	<b>277</b>
8.1. Conclusion . . . . .	277
8.2. Outlook . . . . .	280
<b>Appendix</b>	<b>283</b>
<b>A. Declaration of Published Work</b>	<b>285</b>
<b>B. Problem Definitions</b>	<b>289</b>
<b>C. Algorithms</b>	<b>297</b>
<b>D. Basin Construction</b>	<b>299</b>
<b>E. Experimental Settings</b>	<b>301</b>
<b>F. BEMS Study Documentation</b>	<b>307</b>
<b>G. Statistics</b>	<b>315</b>
G.1. Descriptive Statistics . . . . .	315
G.2. Normal Distribution and Group Differences . . . . .	330
G.3. Interproblem Comparison . . . . .	335
G.4. Performance Rank Tables . . . . .	337
<b>Bibliography</b>	<b>341</b>



# List of Figures

2.1. Illustration of a multi-objective optimization problem (MOOP) and Pareto domination. . . . .	16
2.2. Illustration of preference regions and knee points. . . . .	17
2.3. Comparison of Pareto optimality and Geoffrion proper Pareto optimality. . . . .	22
2.4. Illustration of nondominated sorting. . . . .	26
2.5. Examples of three different cone types. . . . .	27
2.6. Illustration of the tradeoff dominance and preference space. . . .	28
2.7. Decomposition of the tradeoff dominated space using the ordered objectives approach. . . . .	30
2.8. Decomposition of the tradeoff dominated space using the maximum approach. . . . .	33
2.9. Decomposition of the tradeoff dominated space using the minimum approach. . . . .	34
2.10. Illustration of the tradeoff dominated space for extreme values. . .	38
3.1. Illustration of the non-differentiability of the maximal scalarization function. . . . .	43
3.2. Illustration of the ideal, nadir and extreme points. . . . .	47
3.3. Indifference curves of the sum of objectives, product of objectives and Chebyshev method. . . . .	48
3.4. Geometric interpretation of the Nash bargaining solution. . . . .	49
3.5. Illustration of lexicographic extreme points for three objectives. . .	51
3.6. Illustration of the computation of angle utility. . . . .	51
3.7. Illustration of tradeoff utility. . . . .	52
3.8. Example of convex and concave Pareto fronts. . . . .	61
3.9. Illustration of the contraction consistency axiom. . . . .	66
3.10. Illustration of the monotonicity axiom. . . . .	68
3.11. Illustration of the Pigou-Dalton principle. . . . .	71
3.12. Relationship of affine transformations. . . . .	77
4.1. Ontology of preference concepts in multi-objective optimization (MOO). . . . .	85
4.2. Illustration of hypervolume. . . . .	89
4.3. Illustration of solving paradigms in MOO. . . . .	92

List of Figures

4.4. Hierarchy of solving paradigms. . . . . 93

4.5. New solving paradigms in MOO. . . . . 96

4.6. Complete hierarchy of existing and proposed solving paradigms in MOO. . . . . 97

4.7. Working principle of an evolutionary algorithm. . . . . 98

5.1. Illustration of single-point binary crossover. . . . . 114

5.2. Pareto fronts of the problems ZDT1 and ZDT2. . . . . 117

5.3. Pareto fronts of the problem DEB2DK ( $k = 1$  and  $k = 3$ ). . . . . 118

5.4. Pareto fronts of the problem DO2DK. . . . . 119

5.5. Pareto fronts of the two objective Lamé problem ( $\gamma = 0.25$  and  $\gamma = 4$ ). . . . . 121

5.6. Pareto front of the three objective Lamé problem ( $\gamma = 0.5$ ). . . . . 121

5.7. Pareto fronts of the problems DTLZ1, inverted DTLZ1 and DTLZ3. 122

5.8. Pareto fronts of the problems B1, B2 and B3. . . . . 123

5.9. Pareto fronts of the problems B4, B5 and B6. . . . . 124

5.10. Comparison of the expressive power of relative energy (RE) and inverted generational distance (IGD). . . . . 127

5.11. Replacement strategies study – RE. Convergence plots for problems B1 to B6. . . . . 132

5.12. Replacement strategies study – RE. Convergence plots for problems DEB2DK, DO2DK, ZDT1 and ZDT2. . . . . 133

5.13. Replacement strategies study – RE. Convergence plots for problems DTLZ1, invDTLZ1, DTLZ3 and the Lamé problems. . . . . 134

5.14. Replacement strategies – RE. Boxplots. . . . . 135

5.15. Replacement strategies study – IGD. Convergence plots for problems B1 to B6. . . . . 140

5.16. Replacement strategies study – IGD. Convergence plots for problems DEB2DK, DO2DK, ZDT1 and ZDT2. . . . . 141

5.17. Replacement strategies study – IGD. Convergence plots for problems DTLZ1, invDTLZ1, DTLZ3 and the Lamé problems. . . . . 142

5.18. Replacement strategies study – IGD. Boxplots. . . . . 143

5.19. Illustration of crowding distance. . . . . 146

5.20. Pareto fronts of the problems DEB3DK and DTLZ7. . . . . 158

5.21. Pareto fronts of the problems DTLZ5, ZDT3 and ZDT6. . . . . 159

5.22. Comparative study of ESPEA – IGD. Convergence plots for problems B1 to B3. . . . . 165

5.23. Comparative study of ESPEA – IGD. Convergence plots for problems B4 to B6. . . . . 166

5.24. Comparative study of ESPEA – IGD. Convergence plots for problems DEB2DK and DEB3DK. . . . . 167

5.25. Comparative study of ESPEA – IGD. Convergence plots for problems DEB3DK and DO2DK. . . . . 168

5.26. Comparative study of ESPEA – IGD. Convergence plots for problems DTLZ1, invDTLZ1 and DTLZ3. . . . . 169

5.27. Comparative study of ESPEA – IGD. Convergence plots for problems DTLZ5, DTLZ7 and a two objective Lamé problem. . . . . 170

5.28. Comparative study of ESPEA – IGD. Convergence plots for Lamé problems and problem ZDT1. . . . . 171

5.29. Comparative study of ESPEA – IGD. Convergence plots for problems ZDT2, ZDT3 and ZDT6. . . . . 172

5.30. Comparative study of ESPEA – IGD. Convergence plots for convex, concave and mixed curvature problems. . . . . 181

5.31. Example runs of IBEA and MOEAD on the problem DEB2DK ( $k = 10$ ) and a Lamé problem instance ( $\gamma \approx 0.28$ ). . . . . 182

5.32. ESPEA runs on DTLZ3 using the Chebyshev method and on DEB3DK ( $k = 2$ ) using the sum of objectives. . . . . 184

5.33. ESPEA runs on DEB3DK ( $k = 2$ ) using the weighted sum. . . . . 184

5.34. ESPEA runs on DTLZ7 and DEB3DK ( $k = 1$ ) using tradeoff utility. 185

5.35. ESPEA runs on B3 using the Nash bargaining solution. . . . . 186

5.36. ESPEA runs on DEB2DK ( $k = 1$ ) using tradeoff utility and on DO2DK ( $k = 2, s = 1$ ) using the Nash bargaining solution. . . . . 186

5.37. ESPEA runs on DEB2DK ( $k = 3$ ) and DO2DK ( $k = 2, s = 1$ ) using angle utility. . . . . 187

5.38. ESPEA runs on Lamé ( $m = 2, \gamma = 4$ ) with the Chebyshev method and on Lamé ( $m = 2, \gamma = 0.25$ ) with the sum of objectives. . . . . 188

5.39. ESPEA runs on ZDT1 using the Chebyshev method and on ZDT2 using tradeoff utility. . . . . 189

5.40. ESPEA runs on DEB2DK ( $k = 1$ ) using a weighted sum and tradeoff domination. . . . . 190

6.1. Illustration of local scalarization optima and basins. . . . . 193

6.2. Comparison of scalarization landscapes of DO2DK ( $k = 4, s = 1$ ). 204

6.3. Example of a highly multimodal scalarization landscape that makes basin identification impractical. . . . . 205

7.1. Illustration of an artificial neural network (ANN). . . . . 237

7.2. Pareto front of problem instance 2 out of 39 of the cogeneration optimization problem. . . . . 242

7.3. Exemplary search results of ESPEA, IBEA, MOEAD, NSGA-II, NSGA-III, SMPSO, SMS-EMOA and SPEA2 on problem instance 2 of the cogeneration optimization problem for illustrating their performance. . . . . 243

7.4. Surface attainment plot of ESPEA on problem instance 2 of the cogeneration optimization problem. . . . . 244

*List of Figures*

7.5. A single run of ESPEA using fuel efficiency as scalarization function on problem instance 31 of the cogeneration optimization problem. . . . . 246

7.6. Comparison of fuel efficiency between ESPEA and a genetic algorithm (GA). . . . . 247

7.7. Illustration of the encoding of interruptible devices. . . . . 251

7.8. Illustration of the thermal discomfort function. . . . . 258

7.9. Pareto fronts of the problems RW1, RW2 and RW3. . . . . 264

7.10. Pareto fronts of the problems RS1, RS2 and RS3. . . . . 270

7.11. Building energy management system (BEMS) study – illustration of sum of objectives scalarization minima. . . . . 271

7.12. BEMS study – illustration of Nash scalarization minima. . . . . 272

7.13. BEMS study – illustration of angle utility scalarization minima. . . . . 273

7.14. BEMS study – illustration of tradeoff utility scalarization minima. . . . . 274

7.15. BEMS study – illustration of the exhaustive search results in the BEMS study. . . . . 275

F.1. Profiles of outside temperature. . . . . 307

F.2. Solar irradiance. . . . . 308

F.3. Residential building base load. . . . . 308

F.4. Hot water consumption profile of a residential building. . . . . 308

F.5. Load limitation signal. . . . . 309

F.6. Price in cent per kilowatt hour of electricity purchased from the grid. . . . . 309

F.7. Carbon dioxide emissions of consuming a unit of electricity from the grid. . . . . 309

F.8. Load profiles of the operation modes of the washing machine. . . . . 310

F.9. Load profiles of the operation modes and phases of the tumble dryer. . . . . 310

F.10. Load profiles of the operation modes and phases of the dishwasher. . . . . 311

# List of Tables

2.1. Linear inequalities of the ordered objectives approach for three objectives. . . . .	29
2.2. Matrices of the polyhedral cones of the ordered objectives approach for three objectives. . . . .	29
2.3. Matrices of the polyhedral cones of the maximum approach for three objectives. . . . .	32
2.4. Comparison of cone-based decomposition approaches of trade-off optimality. . . . .	36
3.1. Overview of scalarization function parameter dependency. . . . .	45
3.2. Summary of how the presented scalarization functions satisfy the proposed axioms. . . . .	82
5.1. Overview of the time complexity of ESPEA. . . . .	111
5.2. Problems utilized in the replacement strategy study. . . . .	120
5.3. Replacement strategies study – RE. Median and inter-quartile range (IQR) results. . . . .	128
5.4. Replacement strategies study – RE. Anderson-Darling and ANOVA/Kruskal-Wallis test results. . . . .	130
5.5. Replacement strategies study – RE. Post-hoc analysis. . . . .	131
5.6. Replacement strategies study – IGD. Median and IQR results. . . . .	137
5.7. Replacement strategies study – IGD. Anderson-Darling, Bartlett’s and ANOVA/Kruskal-Wallis test results. . . . .	138
5.8. Replacement strategies study – IGD. Post-hoc analysis. . . . .	139
5.9. Additional problems utilized in the comparative study of ESPEA. . . . .	157
5.10. Comparative study of ESPEA – IGD. Median and IQR results. . . . .	173
5.11. Comparative study of ESPEA –IGD. Post-hoc analysis. . . . .	174
5.12. Comparative study of ESPEA – IGD. Friedman test results. . . . .	176
5.13. Comparative study of ESPEA – IGD. Performance comparison by the number of objectives. . . . .	177
5.14. Comparative study of ESPEA – IGD. Median and IQR results grouped by the curvature of the Pareto front. . . . .	178
5.15. Comparative study of ESPEA – IGD. Post-hoc analysis based on the grouping by Pareto front curvature. . . . .	178

*List of Tables*

- 5.16. Comparative study of ESPEA – spread. Median and IQR results grouped by the curvature of the Pareto front. . . . . 179
- 5.17. Comparative study of ESPEA – spread. Post-hoc analysis based on the grouping by Pareto front curvature. . . . . 180
  
- 6.1. Differences between multimodal optimization in single and scalarized multi-objective optimization. . . . . 197
- 6.2. Problems utilized in the study of multimodal preferences. . . . . 203
- 6.3. Basin identification study – useful cluster fraction (UCF) of topographical selection (TS). Median and IQR results. . . . . 211
- 6.4. Basin identification study – detected basin fraction (DBF) of TS. Median and IQR results. . . . . 212
- 6.5. Basin identification study – UCF of nearest better clustering (NBC). Median and IQR results. . . . . 213
- 6.6. Basin identification study – DBF of NBC. Median and IQR results. 214
- 6.7. Local search study – peak distance (PD). Median and IQR results of best local search parametrizations. Median and IQR results. . . 229
- 6.8. Local search study – PD. Post-hoc analysis of best local search parametrizations. . . . . 230
- 6.9. Local search study – PD. Friedman tests results with respect to the baseline. . . . . 231
- 6.10. Local search study – PD. Friedman tests results in detail. . . . . 231
  
- 7.1. Nomenclature used for the cogeneration optimization problem. . 236
- 7.2. Plant components modeled by neural networks. . . . . 238
- 7.3. Mean absolute error (MAE) for training and testing samples for each neural network. . . . . 239
- 7.4. Cogeneration study – IGD. Mean and standard deviation of median results. . . . . 241
- 7.5. Cogeneration study – fuel efficiency. Mean and standard deviation of median results. . . . . 245
- 7.6. Overview of device types in residential buildings. . . . . 249
- 7.7. Overview of the problem instances in the BEMS study. . . . . 259
- 7.8. Overview of heating, ventilation and air-conditioning (HVAC) devices used in the BEMS study. . . . . 260
- 7.9. BEMS study – IGD. Median and IQR results. . . . . 262
- 7.10. Extreme objective values obtained among Pareto optimal points in the BEMS study. . . . . 263
- 7.11. BEMS study – IGD. Post-hoc analysis. . . . . 265
- 7.12. Overview of local preference optima in the BEMS study. . . . . 267
- 7.13. BEMS study – IGD. Median and IQR results of the exhaustive search. . . . . 269

E.1. Experimental settings in the replacement strategies study. . . . . 301

E.2. Experimental settings in the comparative study of ESPEA and the cogeneration study. . . . . 302

E.3. ESPEA configuration in the multimodal preference study. . . . . 303

E.4. Experimental settings in the basin identification study. . . . . 304

E.5. Experimental settings in the local search study. . . . . 304

E.6. Strategy parameters in CMA-ES and their chosen values. . . . . 305

F.1. General parameter overview of the BEMS problem. . . . . 311

F.2. Overview of appliances in summer and winter scenarios in the BEMS study. . . . . 312

F.3. Temporal degree of freedom and standby consumptions of devices. 312

F.4. Technical specification of the air-conditioning. . . . . 312

F.5. Technical specification of the combined heat and power (CHP) plant and the condensing boiler. . . . . 312

F.6. Nomenclature used for the BEMS optimization problem. . . . . 313

G.1. Local search study – PD of CMA-ES. Median and IQR results. . . 316

G.2. Local search study – PD of the GA. Median and IQR results. . . 317

G.3. Local search study – PD of the hill climber (HC). Median and IQR results. . . . . 318

G.4. Local search study – PD of the particle swarm algorithm (PSA). Median and IQR results. . . . . 319

G.5. Local search study – preak ratio (PR) of CMA-ES. Median and IQR results. . . . . 320

G.6. Local search study – PR of the GA. Median and IQR results. . . . 321

G.7. Local search study – PR of the HC. Median and IQR results. . . . 322

G.8. Local search study – PR of the PSA. Median and IQR results. . . . 323

G.9. Local search study – false positives of CMA-ES. Median results. . 324

G.10. Local search study – false positives of the GA. Median results. . . 325

G.11. Local search study – false positives of the HC. Median results. . . 326

G.12. Local search study – false positives of the PSA. Median results. . . 327

G.13. Cogeneration study – IGD. Median and IQR results. . . . . 328

G.14. Cogeneration study – fuel efficiency. Median and IQR results. . . 329

G.15. Comparative study of ESPEA - IGD. Anderson-Darling and Kruskal-Wallis tests results. . . . . 331

G.16. Comparative study of ESPEA – IGD. Anderson-Darling and Kruskal-Wallis test results based on the grouping by Pareto front curvature. 332

G.17. Comparative study of ESPEA – spread. Anderson-Darling and Kruskal-Wallis test results based on the grouping by Pareto front curvature. . . . . 332

G.18. Local search study – PD. Anderson-Darling and ANOVA/Kruskal-Wallis test results. . . . . 333

*List of Tables*

G.19. Cogeneration study – IGD. Anderson-Darling and Kruskal-Wallis test results. . . . .	334
G.20. Cogeneration study – IGD. Post-hoc analysis. . . . .	336
G.21. Performance rank table for the Friedman test in Section 5.3.2. . . .	338
G.22. Performance rank table for the Friedman test in Section 6.3.2. . . .	339

# Glossary

- a posteriori technique** a solving technique in MOO that first obtains an approximation of the entire Pareto front and let the DM choose from this approximation the solution she implements. The Pareto front approximation usually consists of a uniformly distributed discretization of the front.
- a priori technique** a solving technique in MOO that considers the DM's preference before the optimization process is started. Preference information is used to design or configure a tailored algorithm that obtains Pareto optimal solutions that are preferred by the DM.
- black box optimization** a discipline that is concerned with optimization problems of which no information about their algebraic structure is known and only the relation of input and outputs is available.
- box constraints** constraints on the search space of an optimization problem that are given in the form  $\mathbf{x}^l \leq \mathbf{x} \leq \mathbf{x}^r$ , where  $\mathbf{x}^l, \mathbf{x}^r \in \mathbb{R}^n$ . The vectors  $\mathbf{x}^l$  and  $\mathbf{x}^r$  thereby form a box that contains the feasible set  $X$ .
- candidate solution** an element of the feasible set of an optimization problem, i.e. the domain of the objective function.
- codomain** the set of output values into which a function maps. See Definition 2.
- decision vector** an element of the search or decision space of a real-valued MOOP.
- domain** the set of input values on which a function is defined. See Definition 2.
- elitism** an EA is called elitist or suffices elitism, if the individuals that survive to the next iteration are picked from the combined parent and offspring population.

## Glossary

**exploitation** a search strategy during optimization that consists of focusing on a narrow part of the search space in which promising candidate solutions are assumed to be located. Exploitation is equivalent to performing a search on a local scale.

**exploration** a search strategy during optimization that consists of probing large portions of the search space. Exploration is equivalent to performing a search on a global scale.

**function** a binary relation that maps elements of an input set (domain) to an output set (codomain). For each input there exists exactly one output. See Definition 2.

**graph (binary relation)** a collection of ordered pairs defined between elements of a set of departure and elements of a set of destination. The graph of a function describes the functional relationship between inputs and outputs. See Definitions 1 and 2.

**ideal point** the vector that possesses for each entry  $i$  the smallest feasible objective value of objective  $i$  of a given MOOP. See Definition 35.

**implicit function** a mathematical expression of several variables that is equated to 0.

**individual** a candidate solution generated during the execution of an EA.

**interactive technique** a solving technique in MOO that mandates the DM to interact with the optimization algorithm during runtime for steering the search towards regions she deems interesting.

**manifold** a manifold of dimension  $n$  is a topological space that is at each point locally homeomorphic to the  $n$ -dimensional Euclidean vector space. For example, a circle is a manifold of dimension one, since a small segment of the arc locally resembles a line.

**mathematical program** a formulation of an optimization problem that adheres to a predefined algebraic structure. Mathematical optimization algorithms that are designed to solve problems of a specific algebraic structure are guaranteed to find an exact optimum or an optimum within some error bounds of these problems..

**nadir point** the vector that possesses for each entry  $i$  the largest feasible objective value of objective  $i$  among all Pareto optimal objective vectors. See Definition 38.

**normative science** a scientific approach to building theories and models by assuming an idealized state of how a system should function or behave. In MCDA and MOO, a normative approach defines how a decision making process should ideally take place. The DM is expected to adhere to guidelines developed by a normative approach.

**objective vector** an element of the objective space of a real-valued MOOP.

**ontology** a formal system in computer science for naming and defining types, properties and the interrelationships of the entities that are described within the system.

**outranking** a methodology in MCDA that relies on pairwise comparisons to identify preferred alternatives.

**parametric equation** a group of functions of independent variables, where each function can be used to describe a single coordinate of a geometric object.

**plateau** a connected subset of the domain of a function  $f$  on which  $f$  has a constant value.

**population** a set of individuals in an EA.

**positive science** a scientific approach to building theories and models by analyzing factual evidence. In MCDA and MOO, a positivist approach analyzes a DM's choices and behavior to develop a mathematical model that is compatible with his preferences. The generated model is expected to make the same choices as the DM.

**steady state** a property that characterizes EAs that generate only a single new individual in each iteration.



# Acronyms

**ANN** artificial neural network.

**ASF** achievement scalarization function.

**BBO** black box optimization.

**BEMS** building energy management system.

**BFP** best feasible position.

**CHP** combined heat and power.

**CMA-ES** Covariance Matric Adaption Evolutionary Strategy.

**DBF** detected basin fraction.

**DE** differential evolution.

**DM** decision maker.

**EA** evolutionary algorithm.

**ESHL** KIT Energy Smart Home Lab.

**ESPEA** Electrostatic Potential Energy Evolutionary Algorithm.

**FP** false positive.

**GA** genetic algorithm.

**HC** hill climber.

**HVAC** heating, ventilation and air-conditioning.

**IGD** inverted generational distance.

**IP** integer program.

## *Acronyms*

**IQR** inter-quartile range.

**LED** largest energy decrease.

**MaO** many-objective optimization.

**MaOP** many-objective optimization problem.

**MAUT** Multi-attribute Utility Theory.

**MAVT** Multi-attribute Value Theory.

**MCDA** Multiple Criteria Decision Analysis.

**MILP** mixed integer linear program.

**MMO** multimodal optimization.

**MOCO** multi-objective combinatorial optimization.

**MOEA** multi-objective evolutionary algorithm.

**MOO** multi-objective optimization.

**MOOA** multi-objective optimization algorithm.

**MOOP** multi-objective optimization problem.

**NBC** nearest better clustering.

**OWL** Web Ontology Language.

**PD** peak distance.

**PMOEA** preference-based multi-objective evolutionary algorithm.

**PR** preak ratio.

**PSA** particle swarm algorithm.

**PSO** particle swarm optimization.

**PV** photovoltaic.

**RE** relative energy.

**SBX** Simulated Binary Crossover.

**SOO** single-objective optimization.

**SOOP** single-objective optimization problem.

**SOPO** single-objective parametric optimization.

**SWF** social welfare functional.

**TS** topographical selection.

**UCF** useful cluster fraction.

**WIN** worst in archive.



# Notation

The mathematical notation used in this work was chosen to conform with good practices established in textbooks on multi-objective optimization and mathematics. In general, capital latin letters refer to sets (e.g.  $A$ ), boldface capital latin letters to matrices (e.g.  $\mathbf{A}$ ), boldface small latin letters to vectors (e.g.  $\mathbf{a}$ ), latin letters to functions, indices, variables, elements of sets and components of vectors (e.g.  $a$ ). Subscripts (e.g.  $a_i$ ) indicate components of a vector or matrix, superscripts (e.g.  $a^i$ ) represent entities. Due to a limited supply of letters, some symbols are reused throughout multiple chapters and sections with different meaning. Their meaning, however, should be clear in the given context. The following glossary lists symbols that are universally valid throughout the entire thesis.

$d^\perp(\mathbf{u}, \mathbf{v}, \mathbf{w})$  the perpendicular distance of vector  $\mathbf{w}$  to the line passing through the points  $\mathbf{u}$  and  $\mathbf{v}$ .

$<$  a binary relation that establishes a dominance relation. The statement  $a < b$  implies that  $a$  dominates  $b$  according to some dominance notion.

$f|_A$  the restriction of function  $f$  on the set  $A$ , where  $A$  is a subset of the function's original domain  $X$ .

$\text{deg}^-(v)$  the indegree of node  $v$  in a directed graph, i.e. the number of edges that point to  $v$ .

$\text{deg}^+(v)$  the outdegree of node  $v$  in a directed graph, i.e. the number of edges that point away from  $v$ .

$O(f)$  the set of functions that grow asymptotically at most as fast as  $f$ :  
 $\{g : \mathbb{R}_0 \rightarrow \mathbb{R}_+ \mid \exists c > 0 \exists t_0 \forall t > t_0 : f(t) \leq c \cdot g(t)\}$ .

$o(f)$  the set of functions that grow asymptotically strictly slower than  $f$ :  
 $\{g : \mathbb{R}_0 \rightarrow \mathbb{R}_+ \mid \forall c > 0 \exists t_0 \forall t > t_0 : f(t) \leq c \cdot g(t)\}$ .

$w(f)$  the set of functions that grow asymptotically strictly faster than  $f$ :  
 $\{g : \mathbb{R}_0 \rightarrow \mathbb{R}_+ \mid \forall c > 0 \exists t_0 \forall t > t_0 : f(t) \geq c \cdot g(t)\}$ .

## Notation

$\Theta(f)$  the set of functions that grow asymptotically at the same speed as  $f$ :  
 $\{g : \mathbb{R}_0 \rightarrow \mathbb{R}_+ \mid \exists c > 0 \exists t_0 \forall t \geq t_0 : f(t)/c \leq g(t) \leq c \cdot f(t)\}$ .

$\mathcal{L}(S)$  the Lebesgue measure of the set  $S$ , where  $S$  is a subset of a vector space. The Lebesgue measure is a generalization of the volume concept from three dimensions to arbitrary vector spaces.

$M$  the number of units a decision maker (DM) is willing to give up at most in one objective to gain an additional unit in another objective for defining tradeoff optimal solutions (see Definition 14).

$m$  number of objectives of an optimization problem.

$N$  the population size chosen for an evolutionary algorithm (EA).

$n$  number of decision variables of an optimization problem.

$\Phi$  preference predicate. See Definitions 11 and 12.

$\mathbf{x}, \mathbf{y}, \mathbf{z}$  general identifiers for decision vectors.

$\mathbf{u}, \mathbf{v}, \mathbf{w}$  general identifiers for objective vectors.

$\mathbf{u}^{e,k}$  extreme point of the  $k$ -th objective. See Definition 40.

$\mathbf{u}^*$  ideal point. See Definition 35.

$\mathbf{u}^{ndr}$  nadir point. See Definition 38.

$\Psi$  scalarization function. See Definition 28.

$\mathcal{N}(\mu, \sigma)$  a random number drawn from the normal distribution with mean  $\mu$  and standard deviation  $\sigma$ .

$\mathcal{N}_k(\boldsymbol{\mu}, \boldsymbol{\Sigma})$  a vector of random numbers drawn from a multivariate normal distribution of  $k$  components with mean  $\boldsymbol{\mu}$  and covariance matrix  $\boldsymbol{\Sigma}$ .

$\mathcal{U}(a, b)$  a random number drawn from the uniform distribution in the interval  $[a, b]$ .

$[k]$  the set  $\{1, \dots, k\}$ , where  $k \in \mathbb{N}$ .

$\{a\}_{i=j}^k$  the indexed set  $\{a_j, a_{j+1}, \dots, a_{k-1}, a_k\}$ , where  $j, k \in \mathbb{N}$  and  $j < k$ .

$\text{cl}(S)$  the closure of the set  $S$ .

$\text{int}(S)$  the interior of the set  $S$ .

$\text{sort}(S, R)$  sorts the elements in  $S$  according to the binary relation  $R$ , where  $R$  invokes a total order on  $S$ . The sorting  $T = \text{sort}(S, R)$  is a tuple of size  $|S|$  such that for all  $i, j \in [|S|]$  and  $t_i, t_j \in T$  for which  $i < j$  it holds that  $t_i R t_j$ .

$\text{arg sort}(S, R)$  computes a vector of indices  $\mathbf{q} \in \mathbb{N}^{|S|}$  such that for any  $s_i \in S$  the entry  $q_i$  contains the index of  $s_i$  in  $\text{sort}(S, R)$ .

$X$  feasible set of the search space of an optimization problem.

$X_p$  the set of feasible decision vectors that map to Pareto optimal objective vectors.

$\mathcal{Y}$  feasible set in the objective space.

$\mathcal{Y}_p$  Pareto optimal feasible set in the objective space (Pareto front).



# 1. Introduction

Every aspect of our life is affected by choices. From everyday decisions to business propositions or political elections, each choice sets in motion a cascading chain of actions. While some choices are negligible, others impact the fate of entire countries, possibly even mankind. Making good decisions is a difficult task in the face of plentiful options or alternatives that are difficult to fathom. This work aims at providing a better understanding into the preferences that necessarily guide our decision-making process. These insights are used in developing new methods that enable us to make better choices.

This chapter serves as an introduction to the work at hand. The main research question is motivated first. The explanation of concepts from optimization theory, especially multi-objective optimization (MOO), that are discussed therein are based on the fundamental literature of [Deb01, CCLVV07, Mie99, Ehr05, GEF16]. Thereafter, this work is put into perspective of current research on optimization. Section 1.3 discusses the objectives and contributions of this thesis. Finally, the structure of the remaining work is presented in Section 1.4.

## 1.1. Motivation

Optimization is the science of obtaining the best possible outcome by making a choice among a set of different options. In single-objective optimization, the optimization outcome is characterized by only one criterion or objective that represents the outcome's desirability. The optimal choice is obtained by selecting the outcome that fulfills the criterion to the highest degree. Take the example of a firm introducing a new product to the market. Said firm can choose between different production plans, which represent different options, affecting total output and costs per unit produced. Demand for the product depends on the price at which it is offered. Selecting a production plan that maximizes profits is the best possible outcome and desirability is measured in the quantity of money earned.

Outcomes in real-world scenarios, however, are often characterized by multiple criteria, making a clear ranking of alternatives difficult if those criteria are conflicting. Instead of a single solution that optimizes all criteria at the same time,

## 1. Introduction

there exist a set of trade-off solutions that can only be improved in one criterion by deteriorating another criterion at the same time. These solutions are called Pareto optimal or efficient and are all eligible candidates for implementation. MOO is the branch in optimization that deals with such problems. Consider the example of a research and development department of an automotive company designing a new engine. Gasoline consumption and power output are two key properties by which the performance of an engine is judged. Since both criteria are conflicting, there does not exist an engine configuration that minimizes gas consumption and maximizes power output at the same time. Instead, there exist engines that have a high gas consumption and a high power output, ones consuming only little gas exhibiting only little power output and other configurations in between that balance gas consumption and power output.

In most real-world applications, only a single solution or seldom an elected few of the entire Pareto optimal set can be implemented. This circumstance requires a decision maker (DM) to pick only one alternative for implementation. Since there exists no option that is best for all criteria, the *optimal* choice mainly depends on the decision maker's preferences. Going back to the example of the automotive company, the best engine configuration depends on the type of car that is developed. A sports car is expected to have a powerful engine, whereas gas consumption is rather negligible. Compact cars, on the other, should be more economical and favor consumption over power.

Formulating preferences, however, may be a difficult task, since it requires awareness of the underlying values of which the objectives are composed of. Also, DMs are often not aware of their preferences prior to any optimization effort. Preferences sometimes only become apparent when alternatives become available. Aiding the decision maker at any step of the selection process is thereby key in enabling him to make choices that truly reflect the values that form the very basis of her preferences [Kee09]. Consider in this context the example of finding a name for your child. Such a decision has a major impact on the child's future life. At the same time, however, finding a compromise solution between the criteria for judging individual names is tremendously difficult; let alone defining the values that lie at the heart of those criteria.

A convenient way of solving the predicament of choice would be employing a mechanism that ranks all Pareto optimal solutions. The alternative that is placed at the top of the ranking would be considered the optimal choice. Such a ranking mechanism can be composed by eliciting preferences from the DM and employing expert evaluation techniques [GEF16, MA04]. Even if the DM does not have a clear picture about her own preference, expert evaluation techniques can still be used to recommend options that provide a balance between the different objectives.

Most real-world applications measure objectives by real numbers. Aggregating these values into a single real number – a method that is known as *scalarization* – allows establishing an ordering of all Pareto optimal solutions. In that sense, scalarization is a powerful tool that eliminates the necessity of choice. The weighted sum method (see [MA04]) is a well-known example of a scalarization technique. Objective values are multiplied by positive weights signifying their relative importance and subsequently summed. In case the DM does not have any preconceived preferences about the importance of the individual criteria, objective values can at least be normalized to a common scale and summed.

At the same time, however, scalarization bears the risk of oversimplification, since aggregation always results in a loss of information. In the same manner, in which a mathematical problem formulation is only a model of reality, scalarization functions are only a model of human preferences [GEF16]. In both cases, the model might miss unintentionally or willingly certain aspects that form part of reality. The negligence of said aspects is not necessarily the result of an absence of diligence, but the difficulty of finding a mathematical representation of intangible information. Wind farms, for example, are placed such that their power output is maximized and operating costs are minimized. Residents, however, often complain about the placement of individual turbines if they are located close to their homes or seemingly spoil the landscape. Capturing these criteria by a mathematical formulation can be difficult as they are subject to individual perception and, thereby, in general difficult to measure [RM06].

Blindly selecting the optimum identified by the scalarized preference might, therefore, result in a subpar choice. Instead of relying solely on the global scalarization optimum, the scalarization function should serve as a guide to identify alternatives that are interesting to the DM. Since these options are expected to be more desirable, the DM is not overburdened by making a choice between countless options of which most are never even considered for implementation. Narrowing down the choice to interesting alternatives greatly simplifies the decision making process and allows the consideration of hidden information that is not expressed in the scalarization function.

It is therefore imperative that the application of scalarized preferences is amended by optimization techniques that either compute more alternatives to choose from or generate additional information that puts the global scalarization optimum in perspective to other options. The development of such techniques has so far received little attention in MOO. The lack of availability of these techniques motivates a deeper understanding of scalarized preferences and the development of algorithms that provide a DM with more meaningful options to make better choices.

## 1. Introduction

### 1.2. Related Work

Preferences in multi-objective optimization have been a very active field of research in recent years. The field is closely intertwined with the discipline of Multiple Criteria Decision Analysis (MCDA) and both areas distinguish themselves mainly in one key aspect. In MCDA, the set of feasible alternatives to choose from is known to the DM. The problem of identifying the optimal option of these alternatives is called choice problem. In MOO instead, the set of feasible alternatives is unknown prior to any optimization effort. Feasible alternatives must either be obtained by applying some optimization methodology or an optimal choice is generated by the search algorithm itself [BDMS08].

In light of the extensive MCDA and MOO literature, it is imperative that this thesis is put into perspective from both areas of research. Classic MCDA techniques can be broadly divided into two schools. Outranking methods establish an ordering of the different alternatives by pairwise comparisons. Multi-attribute Utility Theory (MAUT) and Multi-attribute Value Theory (MAVT) techniques assign a numeric value to each alternative representing its desirability. MAVT assumes that the outcomes of choices are deterministic, whereas in MAUT, outcomes are probabilistic and utilities are modeled as expected values [GEF16]. For simplicity, and since we only regard deterministic optimization problems, we refer to MAVT in the subsequent work. Scalarization belongs to the family of MAUT/MAVT techniques.

A function within the MAVT framework that assigns numeric values to solutions is called *value function*. Alternatives can be ranked according to the values computed by the value function. Value functions can be further characterized by the degree of additional information that can be derived from the function values. The literature mainly distinguishes between ordinal and cardinal or measurable functions, representing ordinal and cardinal scales of measurement, respectively. Since cardinal preferences are mostly related to utility theory and thereby MAUT techniques, their equivalent in value theory is denoted by measurable preferences [GEF16].

Ordinal values only reveal the position of each option in the ranking and provide no insight into the degree of preference intensity. Measurable functions, on the other hand, allow the derivation of additional knowledge by computing the difference between individual values [GEF16, Roe98]. School grades are an example for an ordinal preference scale. Improving from a 'B' to an 'A', for example, requires a different effort compared to improving from a 'D' to a 'C'. Maximum payload of a spaceship would be an example for cardinal preferences, since increasing the payload from 700 kg to 800 kg is equally beneficial as increasing it from 800 kg to 900 kg.

The analysis in this thesis takes a more general approach to the interpretation of values. We assume that alternatives, whose values are *close* to each other, are somewhat similarly desirable. Strict measurable preferences, which require the comparability of value differences, however, are not enforced. In a practical example, compare a wage increase from 5 € to 10 € to a raise from 1000 € to 1005 €. Although the increase in the amount of money earned is 5 € in both scenarios, the benefit of receiving a 100 % raise in the first case is arguably much higher than the marginal gain in the second case. On the other hand, earning 5 € is more similar to earning 10 € in comparison to 1000 € or 1005 €.

Value functions can be further categorized according to the underlying mathematical model they are based on. The most common approaches in MAVT are based on either the additive or the multiplicative model. In both models, individual criteria are valued by a single-objective value function.<sup>1</sup> The additive model multiplies these valued criteria by individual positive real numbers signifying the relative importance of each objective and sums them up. In the multiplicative model, valued criteria are exponentiated by weights and subsequently multiplied. It is worth noting that in case the valued criteria are positive, any multiplicative value function can be transformed into an additive value function through logarithmizing [GEF16].

The reason, why the additive and multiplicative model are prevalent in MCDA is not just their accessibility. MAVT is based on the axiomatic foundation that a DM possesses a preference structure that can be represented by a value function. In case the preference adheres to a certain set of axioms, there exists an additive or multiplicative value function that represents the preference [GEF16]. Consequently, methods for eliciting weights are a major focus in MAVT research [JLS82].

MOO generally takes a more simplified approach. The valuation of additive and multiplicative value functions greatly depends on the chosen weights [JLS82]. As mentioned before, candidate alternatives are usually not known prior to any optimization effort. This makes the elicitation of preferences difficult, since preferences can change if new information becomes available [KT79]. Therefore, expert knowledge is often sought to construct value functions that guide the search towards solutions that provide a balance between the different objectives [Mie99, MA04, SBS13, DG11, Das99]. These value functions are typically not founded on the basis of an axiomatic framework. The set of Pareto optimal solutions of real-valued optimization problems usually forms a manifold in the objective space called Pareto front. Therefore, value functions and

---

<sup>1</sup>Single-objective value functions can be used to transform qualitative assessments such as ‘good’ or ‘bad’ to a numeric scale. Valuating individual criteria before weighting also allows the modeling of diminishing marginal utility.

## 1. Introduction

preference notions in general are often inspired by its geometric properties [BDDO04, DG11, BSS17, Das99, SW13].

So far, we have implicitly assumed that the valuation of individual criteria are independent of each other. Extended MAVT models, however, also consider interactions between criteria. Redundancy and complementarity between objectives can be modeled using the discrete Choquet integral [Cho54]. Interactions between criteria are accommodated in the additive model by giving weights to subsets of criteria instead of individual criteria. An in-depth explanation of this methodology is beyond the scope of this work. The interested reader, however, is referred to [Gra96] for a detailed description.

Synergies between the valuation of objectives are usually not explicitly considered in classic MOO techniques. Instead, positive or negative interdependencies are directly handled at the objective level, since they affect the composition of the Pareto optimal set. If two objectives are aligned in the sense that the improvement of the first objective always results in an improvement of the second objective and vice-versa, the dimensionality of the Pareto front is reduced by one, since there exists no tradeoff between both objectives [Pad13]. A Pareto optimal solution can be obtained by only optimizing one of the two objectives. Objective function evaluations of real-world optimization problems are usually computationally expensive [Kno06, Som81, SCBS14]. Therefore, a case can be made for omitting redundant or strongly correlated objectives to save computation resources [DS05]. Of course, redundant objectives should be considered in the final decision making process as each criterion influences the usefulness of a solution to implement.

In this thesis, scalarization broadly refers to any function that aggregates a vector of numeric values to a single real value. We do not require any axiomatic system of preferences to guarantee the existence of such functions. We do assume, however, that said functions can be characterized by certain mathematical properties that guarantee that a function behaves in a certain way. The discussion of these properties plays a central part in Chapter 3.

### 1.3. Objectives and Contributions

This work addresses the role of scalarization as a method for formulating preferences in MOO. The main goal of this thesis is, firstly, providing a better understanding of scalarized preferences in MOO and, secondly, to develop new methods that use scalarized preference information to generate more solutions that are interesting to the DM. The contributions of this thesis can be summarized in the following way:

- the proposition of a new framework for the mathematical characterization of preferences in MOO and the description of scalarized preferences in said framework,
- the identification and definition of mathematical properties scalarization functions can exhibit that invoke desirable behavior, which is beneficial for the choice and optimization process,
- an algorithmic characterization of preferences that is based on the optimization result delivered to the DM,
- the development of a new methodology for describing and obtaining optimal Pareto front approximations by a finite set of points biased by scalarized preference information,
- the development of a new technique for obtaining local scalarized preference optima,
- a computational study of the proposed algorithms on artificial benchmark problems and in real-world application scenarios.

For a summary of previous publications that have contributed to this work, see Appendix A.

## 1.4. Structure

The remaining work is structured as follows. The next chapter provides a general introduction to preferences in multi-objective optimization. The multi-objective problem is motivated as an extension to the single-objective problem. Preferences are introduced in a two-fold way; firstly as a restriction to the set of Pareto optimal solutions and secondly as a transformation of the original optimization problem. Three distinctive transformation approaches are identified and illustrated using a practical example of a tradeoff-based preference model.

The subsequent chapter formally introduces the notion of scalarized preferences and categorizes them in the framework presented in the previous chapter. The notion of a scalarization function to express preferences is formally defined. Specific instances of scalarization functions are discussed and categorized. Subsequently, mathematical properties that a scalarization function may exhibit are presented. The fulfillment of such properties implies that the scalarization function behaves in a certain way, which in turn may be desired by a DM. It is shown, under which conditions the presented scalarization function fulfill the proposed properties.

## 1. Introduction

Chapter 4 presents an algorithmic characterization of preferences in MOO. A framework is presented for characterizing the methodology of optimization with respect to the optimization outcome and, therein, different paradigms are described. Two new paradigms are developed in the course of said framework – preference-biased Pareto front approximations and local preference optima. The chapter also provides a brief introduction to evolutionary algorithms and a justification to why this approach is chosen to develop algorithms that implement the new paradigms.

Next, a new concept for defining an optimal distribution of points on the Pareto front using scalarized preference information is introduced. The theoretical properties of this concept is examined and it is shown that such distributions exist under very mild conditions. An algorithm – the Electrostatic Potential Energy Evolutionary Algorithm (ESPEA) – that implements this concept to generate preference-biased Pareto front approximations is presented. Different selection strategies for choosing candidate solutions in ESPEA are discussed and computationally analyzed. ESPEA is compared to other state-of-the-art multi-objective evolutionary algorithm (MOEA) to examine how well ESPEA performs in obtaining uniform Pareto front approximations if all Pareto optimal points are considered to be equally desirable. A qualitative analysis of the preference-biased Pareto front approximations obtained by ESPEA examines their usefulness from a decision-making perspective.

An algorithmic framework for obtaining local scalarized preference optima is presented in Chapter 6. The framework consists of three components. Different methodologies for the individual components are developed and examined. Quantitative computational studies are performed to compare the methodologies. The computational study puts its focus on finding optimal parametrizations for each method and finding optimal methods for each component.

Chapter 7 presents two real-world applications for the algorithms developed in Chapters 5 and 6. The first application is concerned with the optimization of the operation of a combined heat and power (CHP) plant. It is shown that ESPEA is able to obtain a uniform approximation of the Pareto front of the considered problem. ESPEA is also able to generate preference-biased Pareto front approximations that contain more solutions that maximize the plant efficiency while retaining a scope of the entire Pareto front. The second application is concerned with finding optimal schedules for devices in a residential building. Minimizing energy consumption costs, carbon dioxide emissions and thermal discomfort are considered as objectives in this context. It is shown that schedules that are local preference optima present themselves as interesting options to the DM. At the same time, restricting the optimization result to local scalarization optima simplifies the decision making process. The thesis is concluded by a summary and an outlook on future work.

## 2. Preferences in Multi-objective Optimization

The fundamentals of preferences in MOO are discussed in this chapter. Optimization problems and preferences are defined on the basis of set theory and the theory of relations in contrast to existing approaches that rely on numeric representations of variables and objectives [Deb01, CCLVV07, Mie99, Ehr05]. A set- and relation-based focus allows a clear and coherent definition of preferences in MOO that is independent of the algebraic structure of the problem to be solved. Thereby, existing concepts related to preferences in MOO can be generalized and extended beyond real-valued vector spaces. The proposed characterization of preferences is a first step towards developing optimization methodologies that are applicable to arbitrary algebraic structures.

This chapter starts by explaining the foundations of optimization theory. The multi-objective optimization problem (MOOP) is introduced as an extension to the single-objective optimization problems (SOOPs) in optimizing multiple objectives at the same time. Pareto optimality is presented as the combination of the orderings that are imposed on the individual objectives of the SOOPs. Preferences can then be described as a restriction to the set of Pareto optimal solutions. The second part of the chapter focuses on how preferences can be articulated as a transformation of the original MOOP and how different approaches to preference formulations can be related to each other. The chapter is concluded with the presentation of three distinct formulations of a tradeoff-based preference notion that identify the same preferred subset of the set of Pareto optimal solutions.

### 2.1. Multi-objective Optimization Problems

Broadly speaking, any optimization problem consists of a set of inputs, a set of outputs, and a mapping that relates elements of both sets to each other. In the example of Section 1.1, a firm can choose between different production plans for maximizing its profit from product sales. In this context, the different production plans are inputs, profits are outputs, and the mapping describes how

## 2. Preferences in Multi-objective Optimization

production plans are related to outputs. More formally, the mapping arranges inputs and outputs as ordered pairs. Let us assume that the firm can choose between a high capacity and a low capacity production plan, which in turn result in large and small profits. Then, the set of inputs is {high, low} and the set of outputs is {large, small}. If high capacity results in large profits and low capacity in small profits, the mapping consists of the ordered pairs {(high, large), (low, small)}. Binary relations are a concept in mathematics that can be used to describe these input-output relationships (see Definition 1).

**Definition 1** (binary relation [DM60]). *A binary relation  $R$  is a triple  $(A, B, G)$ , where  $A$  and  $B$  are sets and  $G$  is a subset of the Cartesian product  $A \times B$ . Furthermore,*

- $A$  is called the set of departure,
- $B$  is called the set of destination,
- $G$  is called the graph of  $R$  and
- for any  $a \in A$  and  $b \in B$  the statement  $(a, b) \in G$  or  $aRb$  reads that  $a$  is  $R$ -related to  $b$ .

In an optimization context, input-output relations are characterized by each input being mapped onto exactly one output requiring a restriction of the binary relation concept. Different inputs may result in the same output, however no input may lead to multiple outputs. In the example of the firm, two production plans may result in the same profit margin, however a single production plan cannot yield two profit levels. A binary relation as of Definition 1 would allow such pairings. Optimization problems thereby require a restricted type of binary relation that guarantees, on the one hand, that there exists an output for every input and, on the other hand, every input is mapped onto only one output. The function (see Definition 2) is a binary relation that satisfies these requirements.

**Definition 2** (function [Bou54]). *A function  $f$  is a binary relation  $(X, Y, F)$  such that for all  $x \in X$  there exists exactly one  $y$  such that  $(x, y) \in F$ . Furthermore,*

- $X$  is called the domain of  $f$ ,
- $Y$  is called the codomain of  $f$ ,
- $\mathcal{Y} := \{y \in Y \mid \exists x \in X : f(x) = y\}$  is called the image of  $f$  or the feasible set of the codomain,
- $f(x) = y$  states that  $x$  maps to  $y$ , where  $y$  is the image of  $x$  and  $x$  is a preimage of  $y$ .

## 2.1. Multi-objective Optimization Problems

In real-world applications, every output is usually influenced by multiple inputs. In the example of the firm, a production plan may consist of the design of a manufacturing plant. The capacity of a bottling plant, for example, depends on the capacity of the bottling machines and packaging. The size of both components can be chosen separately in designing the plant. A single element of the domain of a function can, thereby, itself be comprised of multiple values. This specific type of function is referred to as multivariate function (see Definition 3). In the following, we do not explicitly distinguish between functions as of Definition 2 and multivariate functions, since any distinction does not affect the subsequent analysis.

**Definition 3** (multivariate function [Bou54]). *A function  $f = (X, Y, F)$  is called multivariate iff  $X = X_1 \times \dots \times X_n$  with  $n \geq 2$  and the  $X_i$  for all  $i \in [n] := \{1, \dots, n\}$  are non-empty sets.*

The function builds the basis of any optimization problem. Single-objective optimization (SOO) seeks to identify the *best* attainable value in  $Y$  – the optimum – and an element of  $X$  that maps to the best value. The firm is interested in obtaining the highest attainable profits and a production plan through which these profits can be realized. This requires that elements of  $\mathcal{Y}$  can be compared in such a manner that there exists an output that is either better or at least as good as all other options. In practice, comparability is usually inherent to the objective that is optimized. For example, it is quite obvious that a firm always prefers larger profits. In a formal model, an order is imposed on  $\mathcal{Y}$  by a binary relation  $R$  that allows a ranking of the elements of the image of  $f$ . If an optimum exists, any such binary relation must define a least element (see Definition 4)<sup>1</sup>, which is the optimum that is sought.

An SOOP is only deemed *solvable* if a least element exists. In case a least element does not exist, it might still be possible to obtain an element of  $\mathcal{Y}$  that is a satisfying output.<sup>2</sup> Such considerations, however, go beyond the scope of this formal model. We consider only solvable problems in this work. In SOO, it is commonly assumed that all elements of  $\mathcal{Y}$  can be compared to each other, that the best attainable value is unique and that there exist no cyclic relations. We adopt these assumptions, since they are useful and reasonable from a practical perspective. A binary relation that defines a weak wellorder fulfills all these requirements (see Definition 4).

**Definition 4** (weak wellorder cf. [Can82]). *A binary relation  $R = (A, A, G)$  is called*

---

<sup>1</sup>A greatest element can be defined equivalently.

<sup>2</sup>The real line  $(0, 1)$ , for example, does not possess a least element for the  $\leq$ -relation, since for any number close to 0 there can be found a smaller number that is even closer to 0. However, any number that is sufficiently close to 0 could be deemed acceptable from a decision making point of view.

## 2. Preferences in Multi-objective Optimization

- complete if  $\forall a, b \in A$  either  $aRb$  or  $bRa$ ,
- antisymmetric if  $\forall a, b \in A$   $aRb$  and  $bRa$  implies  $a = b$ ,
- transitive if  $\forall a, b, c \in A$   $aRb$  and  $bRc$  implies  $aRc$  and
- possesses a least element if  $\exists a \in A$  such that  $\forall b \in A$   $aRb$  holds.

A weak wellorder is a binary relation on a set  $A$  that is complete, antisymmetric, transitive and defines a least element on  $A$ .<sup>3</sup>

The least element is *at least as good* as any other available option. Completeness asserts that all elements of  $\mathcal{Y}$  are comparable to each other and thereby guarantees that the optimum can be compared to all alternatives. Antisymmetry states that if  $a$  is at least as good as  $b$  and vice versa,  $a$  and  $b$  necessarily coincide. This implies in the example of the firm that large profits are at least as good as large profits. Antisymmetry guarantees that the optimal value is unique. Transitivity prevents the occurrence of preference cycles. The existence of a least element finally asserts that an optimal value exists. Using the concept of the weak wellorder, the SOOP can be defined as in Definition 5.

**Definition 5** (single-objective optimization problem (SOOP)). *Let  $f = (X, Y, F)$  be a function, where the binary relation  $R$  imposes a weak wellorder on the image  $\mathcal{Y}$  of  $f$ . The single-objective optimization problem consists of obtaining an element  $x^* \in X$  that maps to the least element of  $\mathcal{Y}$ . Furthermore,*

- $X$  is called the feasible set of  $f$ ,
- any  $x \in X$  is called decision variable. If  $f$  is multivariate, the  $x_1, \dots, x_n$  are called decision variables,
- any  $y \in Y$  is called objective value,
- $x^*$  is called optimizer or solution of  $f$ ,
- $y^* := f(x^*)$  is called optimal value of  $f$  and
- $f$  is called objective function.

An MOOP could be understood as finding a solution to multiple SOOPs at the same time. Conversely, an SOOP can be considered a special case of an MOOP that has only one objective function. The underlying objective functions of the SOOPs are combined into a joint function that forms the basis of the MOOP. The construction of this function requires the joining of the individual

---

<sup>3</sup>Cantor requires the wellorder to possess a least element for any non-empty subset of  $A$ . This prerequisite is too strong for the presented model, since neither the  $\leq$ - nor the  $<$ -relation define a least element on all subsets of  $\mathbb{R}$ . The weak wellorder in this thesis is not related to the weak wellorder concept in [Flu13].

## 2.1. Multi-objective Optimization Problems

domains, codomains and function graphs. Joining domains implies defining a common domain for all objective functions by combining inputs of the individual domains such that each combined input of the common domain yields outputs that are feasible for all objective functions. Constructing such a domain from domains that are composed of different inputs is difficult, since individual problems might impose different restrictions on the set of available inputs. Budgets, for example, restrict available resources that can be divided between inputs. The firm that plans to release a new product might also consider the environmental impact of its production as secondary goal. Investing in technology to prevent pollution might lock resources that could be used to expand production instead. Since the joining of individual domains does not yield any greater insight into preferences in MOO, we assume that all the objective functions that are joined are defined on the same domain. This assumption is common throughout the scientific literature [Deb01, CCLVV07, Mie99, Ehr05].

The joint codomain of multiple SOOPs that are combined into an MOOP can be modeled as the Cartesian product of the images of the individual functions. Elements of the multi-objective codomain are tuples that possess as many entries as there are objectives. Finally, the joint function graph consists of pairs, where the first entry is an element of the joint domain and the second entry is an element of the Cartesian product of the images of the objective functions. Secondly, for each member of the joint graph there must exist a mapping from the domain to every individual image (see 6).

**Definition 6** (multi-objective function). *For an  $m$ -tuple of SOOPs let  $f := (f_1, \dots, f_m)$  be the  $m$ -tuple of corresponding objective functions with for all  $i \in [m] : f_i = (X, Y_i, F_i)$ . The multi-objective function is given by  $(X, Y, F)$ , where*

- $Y := \mathcal{Y}_1 \times \dots \times \mathcal{Y}_m$  and
- $F := \{(x, y) \in (X \times Y) \mid y = (y_1, \dots, y_m) \wedge \forall i \in [m] \Rightarrow \exists x \in X : f_i(x) = y_i\}$ .

The multi-objective function builds the basis of the MOOP. The next step consists of formally defining the MOOP by stating which elements are sought in MOO. When simultaneously searching for an optimum of multiple conflicting objectives that share common inputs, we cannot expect that there exists an element in  $X$  that maps to each optimal value of every SOOP. Evidently, maximizing production capacity must come at the cost of environmental impact. Conversely, minimizing environmental impact always results in a loss of production capacity. The weak wellordering of each individual image, however, allows inferring a consecutive order of the codomain of the multi-objective function. An element of the joint codomain is *better* compared to another alternative if it is at least as good in every objective and strictly better in at least one objective. This order is called Pareto order (see Definition 7).

## 2. Preferences in Multi-objective Optimization

**Definition 7** (Pareto order cf. [Par96]). Let  $A := \{(a_1, \dots, a_m) \mid a_i \in A_i\}$ , where all  $A_i$  are weak wellordered sets imposed by the binary relation  $<_i$ . The Pareto order  $<_p$  is a binary relation on  $A$  that is defined as follows: for all  $a, b \in A$  it holds that  $a <_p b$  iff  $a_i <_i b_i$  for at least one  $i \in [m]$  and at the same time  $\nexists j \in [m]$  such that  $b_j <_j a_j$ . Furthermore, for  $a, b \in A$

- $a <_p b$  reads  $a$  Pareto dominates  $b$ ,
- if  $a \not<_p b$  and  $b \not<_p a$  then  $a$  and  $b$  are non-dominated to each other and
- if  $\nexists c \in A$  such that  $c <_p a$ , then  $a$  is called Pareto optimal or a minimal element of  $A$ .

The Pareto order provides a means of ranking elements of the image of a multi-objective function. However, it does not relate all elements of the joint image to one another, as there exist elements for which neither dominates the other. The minimal elements of the image of a multi-objective map – those that are not Pareto-dominated by any other element – are optimal in the sense that the ordering of the original SOOPs, from which they stem, is preserved. No element that is worse in one objective can Pareto dominate another element that is better in said objective. In case that there do not exist any preferences regarding the individual objectives, any such minimal element constitutes a viable candidate for implementation as it can only be improved in one objective by deteriorating another objective at the same time. The Pareto optimal elements are therefore the set of solutions to an MOOP (see Definition 8).

**Definition 8** (multi-objective optimization problem (MOOP)). Let  $f = (X, Y, F)$  be a multi-objective function. The multi-objective optimization problem consists of obtaining a preimage of every Pareto optimal element of  $\mathcal{Y}$ . Furthermore,

- $\mathcal{Y}_p := \{y \in \mathcal{Y} \mid \nexists \bar{y} \in \mathcal{Y} : \bar{y} <_p y\}$  is called the Pareto front,
- $X_p := \{x \in X \mid f(x) = y \in \mathcal{Y}_p\}$  is called set of Pareto optimal solutions and
- any  $x \in X_p$  is called a Pareto optimal solution.

So far, an abstract perspective has been kept on the topic of optimization in defining the domain and codomain of a problem as sets of arbitrary elements. That perspective is a necessary requirement for providing a rigorous mathematical foundation of preferences in MOOP. Nearly all real-world applications and artificial benchmark problems, however, feature a real-valued vector space as codomain. Decision variables are usually real-valued themselves, integers, binary values or a combination of each. For developing efficient procedures to obtaining Pareto optimal solutions, it is imperative that the domain and codomain exhibit some meaningful algebraic structure to avoid brute force approaches that enumerate the entire domain. Scalarization itself is a technique

## 2.1. Multi-objective Optimization Problems

that requires objectives to be measured on a real-valued scale. The real-valued MOOP that serves as the basis of the analysis in subsequent chapters is therefore given in Definition 9 and illustrated in Figure 2.1a.

**Definition 9** (real-valued MOOP cf. [CCLVV07]). *Let  $f = (X, Y, F)$  be the objective function of an MOOP. The MOOP is called real-valued if  $X \subseteq \mathbb{R}^n$  and  $Y \subseteq \mathbb{R}^m$ , where  $n, m \in \mathbb{N}$  and  $m \geq 2$ . Additionally, for all  $i \in [m]$  each  $Y_i$  is ordered by the  $\leq$ -relation. Furthermore,*

- $\mathbb{R}^n$  is called the search space of  $f$ ,
- any  $\mathbf{x} \in \mathbb{R}^n$  is called decision vector,
- $\mathbb{R}^m$  is called the objective space of  $f$ ,
- any  $\mathbf{u} \in \mathbb{R}^m$  is called objective vector and
- $\mathcal{Y}$  is called the feasible set of the objective space,

Elements of vector spaces are referred to as points or vectors. For simplicity and clarity, we consider only minimization problems in the analysis in subsequent chapters. This restriction does not present a limitation, however, since any maximization problem can be transformed into an equivalent minimization problem by negating its objective functions [Ber99].<sup>4</sup> The domain of a real-valued MOOP is defined by imposing constraints on the search space that are expressed as either equalities or inequalities. Thereby, the values that each decision variable is allowed to take on, is restricted. Pareto domination in vector spaces is formally given in Definition 10 and illustrated in Figure 2.1b. We use the common notation of writing elements of vector spaces and matrices in **bold face letters** (see for example Definition 10).

**Definition 10** (Pareto order in vector spaces [Par96]). *Let  $\mathbf{y}^1, \mathbf{y}^2 \in \mathbb{R}^m$ . The expression  $\mathbf{y}^1 <_p \mathbf{y}^2$  is equivalent to for all  $i \in [m]$  it holds that  $y_i^1 \leq y_i^2$  and there exists a  $j \in [m]$  such that  $y_j^1 < y_j^2$ .*

In the next section, preferences in MOO are formally introduced. To guarantee that the discussion and its results apply to all MOOP classes, the general MOOP as of Definition 8 is considered instead of its real-valued counterpart of Definition 9.

---

<sup>4</sup>Maximization problems correspondingly require the existence of a greatest instead of a least element.

## 2. Preferences in Multi-objective Optimization

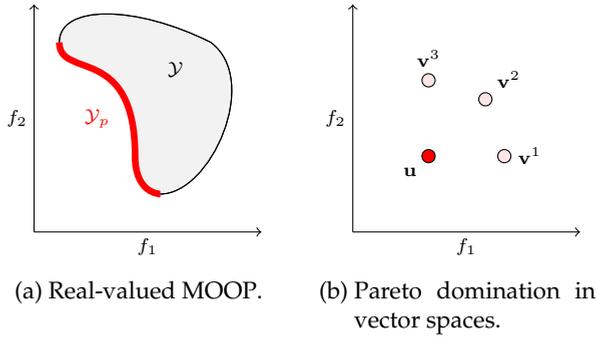


Figure 2.1.: Illustration of a bi-objective MOOP and Pareto domination in a two-dimensional vector space. Figure 2.1a shows the feasible set of the objective space  $\mathcal{Y}$ . The Pareto front is a continuous curve colored in red. The vector  $\mathbf{u}$  dominates the points  $\mathbf{v}^1$ ,  $\mathbf{v}^2$  and  $\mathbf{v}^3$  in Figure 2.1b.

## 2.2. Preference Modeling

Since the inception of MOO as a discipline of research, preferences have been a subject of study in the field. Every MOOP requires a DM to make a choice about the solution that is finally implemented. Choosing a solution for implementation always involves making a compromise between different objectives [GEF16]. Preferences are often regarded from a practical perspective as to how a DM can be enabled to arrive at her favored solution. For this reason, frameworks that categorize preferences mostly focus on how preferences are or can be implemented in multi-objective optimization algorithms (MOOAs) (see for example [LYBF<sup>+</sup>16, Coe00, RS06]). A discussion of these taxonomies is given in Section 4.1.

Preferences, however, should not only be analyzed from an algorithmic perspective, but also on the problem level itself. As stated in the previous section, Pareto optimality is only a natural extension of the order that is imposed on individual objectives. If no assumptions about the DM's preferences can be made, any Pareto optimal solution is an equally good candidate for implementation. Ranking or preferring some solutions to others thereby restricts the set of solutions that are deemed *optimal*. In that sense, the original MOOP can be transformed such that the preferred set of alternatives is the solution to a modified MOOP that incorporates the DM's preferences.

There exist few frameworks in the literature that have taken the approach of describing preferences as a problem transformation. Li et al. have proposed an

## 2.2. Preference Modeling

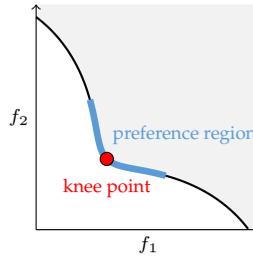


Figure 2.2.: Illustration of preference regions and knee points. A preference region is a subset of the Pareto front and usually located at convex bulges of the Pareto front. Elements of a preference region are preferred by the DM to other Pareto optimal points that are not a member of that region. A knee point is a point that is located at the center of a convex bulge of the Pareto front.

ontology for categorizing preference-based multi-objective evolutionary algorithms (PMOEAs) [LYBF<sup>+</sup>16, LYBF<sup>+</sup>17] as well as *preference models*. Preference models refer to the methodology that is used in obtaining preferred solutions. Its instances include well known concepts from MOO such as knee points or preference regions (see Figure 2.2), but also from classic MCDA techniques, which are traditionally well described and categorized. The ontology acknowledges the existence of PMOEAs that perform objective space transformations to obtain preferred solutions, however their analysis focuses on an algorithmic description of preferences and does not propose a formalism to express preferences as problem transformations.

Shukla et al. distinguish between direction- or region-based models and slope- or curvature-based models [SED13]. In direction- and region-based models, the DM's preference guides the search towards areas of the Pareto front she deems interesting. In slope- and curvature-based models, the desirability of a point depends on the composition of the Pareto front, e.g. the curvature of the curve that represents the Pareto front in bi-objective problems. Shukla et al. describe preference models as triples of objective function – which they conceive as combination of function graph and codomain – function domain and an order imposed on the objective space. Any such triplet defines a preferred subset of the set of Pareto optimal solutions. Their methodology acknowledges preferences as modification to the original optimization problem.

In this thesis, a preference is a choice rule that defines a preferred subset of the set of Pareto optimal solutions. Such choice rules can be modeled as predicates in mathematical logic and are foremost independent of the actual problem structure (see Definition 11). Note that  $X_p$  does not have to be known to the DM for him being able to define a preference. In this sense, preferences

## 2. Preferences in Multi-objective Optimization

must be understood as a mechanism, whose invocation on  $X_p$  defines a smaller subset without knowing the composition of  $X_p$ . For a practical example, see Section 2.3.

**Definition 11** (preferred subset). *A preference  $\Phi$  is a predicate that defines a preferred subset  $\mathcal{X}_p$  of  $X_p$ :*

$$\mathcal{X}_p := \{x \in X_p \mid \Phi(x)\}. \quad (2.1)$$

Definition 11 provides only an abstract concept to formally characterize preferences. As stated before, preferences should be integrated into the problem formulation such that the preferred subset is the solution to a transformed problem. An MOOP is defined by its objective function and the orderings of its individual objectives, and the objective function can be decomposed into domain  $X$ , codomain  $Y$  and graph  $F$ . Any modification must therefore induce a change in these components. Of course, these changes cannot be arbitrary and must preserve certain structures of the original problem. Otherwise, the modified problem might be no longer a representation of the physical reality on which the original problem is based. In the following, each component is discussed individually, before a general model is presented.

**Domain** The domain of a function is a set that can be transformed by changing the composition of its elements. Extending the domain or replacing its members by elements of other sets could allow the consideration of infeasible inputs. This would no longer constitute a valid transformation, but a definition of an entirely new optimization problem. Functions are characterized by the property of *left-totality*, which states that for every element of the domain, there must exist a map to the codomain [Bou54]. By introducing new elements to the domain and thereby necessarily new mappings, extension and replacement could also lead to new elements entering the Pareto optimal set, which would contradict Definition 11. Therefore, the only viable option of transforming the domain is to remove elements from it, thereby choosing a subset of the original domain. Real-valued search spaces can be cropped, for example, by imposing additional equality or inequality constraints.

**Codomain** Describing valid and meaningful transformations of the codomain is more difficult. Adding elements to the codomain has obviously no effect as there exists no member of the domain that map to them. Removing elements from the codomain does not change the composition of  $X_p$  if they do not form part of the image. Eliminating a member of the image, on the other hand, violates left-totality. A viable and meaningful transformation can thereby only consist of mapping the image itself to a different codomain  $Z$ . By concatenating the mapping from  $X$  to  $Y$  and  $\mathcal{Y}$  to  $Z$  we can effectively replace  $Y$  with  $Z$  in the objective function. Such an approach is, for example, taken in MAVT. By aggregating objective values, objectives are mapped to the field

of real numbers. Elements of  $Z$  can also themselves be composed of tuples such as elements in  $Y$ . It is only mandatory that there exists an order on the elements in  $Z$ .

**Graph** Any domain and codomain transformation implicitly changes the graph of the objective function. Manipulating the graph only would correspond to changing the relation between inputs and outputs. Although such an approach could be used to pair Pareto optimal solutions with dominated elements of the image to remove members from the Pareto set, it makes more sense to combine a graph transformation with a transformation of the domain or codomain.

**Order** Changing the binary relation that orders the codomain is the most straightforward approach for formulating preferences in MOO. Outranking techniques, for example, fall in this category. Any binary relation  $R$  that is imposed as a preference on  $Y$ , however, must be an extension of the Pareto order. Otherwise, rank reversals could occur that violate the natural order of the individual objectives and Pareto dominated solutions might enter the preferred set. In case, the preference encompasses a codomain transformation as well,  $R$  is directly imposed on  $Z$ .

Combining the results of the analysis, the mathematical structure of preference predicates is given in Definition 12. Note that Definition 12 explicitly requires the transformed domain  $X_\Phi$  to be a subset of the original domain  $X$  and that the preferred order  $<$  does not violate the Pareto order.

**Definition 12** (preference predicate). *Let  $f = (X, Y, F)$  be the objective function of an MOOP. A preference predicate  $\Phi = (f_\Phi, <)$  is a pair of a function  $f_\Phi = (X_\Phi, Y_\Phi, F_\Phi)$  with  $X_\Phi \subseteq X$  and a binary relation  $<$  that orders  $Y_\Phi$ . Additionally, for all  $x^1, x^2 \in X_\Phi$  for which  $f(x^1) <_p f(x^2)$  it holds that  $f_\Phi(x^1) < f_\Phi(x^2)$ . Then, for any  $x \in X$  the expression  $\Phi(x)$  is true iff*

- $x \in X_\Phi$  and
- $\forall x^* \in X_\Phi$  for which  $f_\Phi(x^*) < f_\Phi(x)$  it holds that  $f_\Phi(x^*) = f_\Phi(x)$ .

According to Definition 12, any preferred element  $x$  is required to fulfill two conditions. Firstly,  $x$  has to be a member of the transformed function domain  $X_\Phi$ . Secondly, the image of  $x$  must not be dominated by any element  $Y_\Phi$ , where domination is induced by  $<$ . Definition 12 shows that there exist different methodologies of stating preferences in MOO. Theorem 1 shows that these approaches can be transformed into each other.

**Theorem 1.** *Let  $f = (X, Y, F)$  be the objective function of an MOOP. For every preference predicate  $\Phi = ((X_\Phi, Y_\Phi, F_\Phi), <)$  there exist corresponding preference predicates  $\Phi_X = ((X_X, Y, F_X), <_p)$ ,  $\Phi_Y = ((X, Y_Y, F_Y), <_p)$  and  $\Phi_{<} = ((X, Y, F), <_{<})$*

## 2. Preferences in Multi-objective Optimization

) such that for all  $x \in X_p$

$$\Phi(x) = \Phi_X(x) = \Phi_Y(x) = \Phi_{<}(x).$$

*Proof.* The preferred sets of  $\Phi$ ,  $\Phi_X$ ,  $\Phi_Y$  and  $\Phi_{<}$  are given by  $\mathcal{X}_p^s = \{x \in X_p \mid \Phi_s(x)\}$  where  $s \in \{\emptyset, X, Y, <\}$ . For the proof of Theorem 1, it is sufficient to show that given  $\Phi$  it is possible to construct predicates  $\Phi_X$ ,  $\Phi_Y$  and  $\Phi_{<}$  for an arbitrary  $f$  such that  $\mathcal{X}_p = \mathcal{X}_p^X = \mathcal{X}_p^Y = \mathcal{X}_p^{<}$ .

1. Let  $\Phi_X = (f_X, <_p)$  with  $f_X = (\mathcal{X}_p, Y, F_X)$  and  $F_X = \{(x, y) \in F \mid x \in \mathcal{X}_p\}$ . For all  $x, x^* \in \mathcal{X}_p$  for which  $\Phi_X(x)$  and  $f_X(x^*) <_p f_X(x)$  it follows that  $f_X(x^*) = f_X(x)$ , since by requirement of Definition 12 both  $f_X(x^*)$  and  $f_X(x)$  are Pareto optimal in  $f_X(\mathcal{X}_p)$ . It follows that  $\mathcal{X}_p = \mathcal{X}_p^X$ .
2. Let  $\Phi_Y = (f_Y, <_p)$  with  $f_Y = (X, Y_Y, F_Y)$ ,  $Y_Y = \{0, 1\}$  and  $F_Y = \{(x^0, 0), (x^1, 1) \in X \times \{0, 1\} \mid x^0 \in \mathcal{X}_p \text{ and } x^1 \in X \setminus \mathcal{X}_p\}$ . For all  $x, x^* \in X_p$  for which  $\Phi_Y(x)$  and  $f_Y(x^*) <_p \Phi_Y(x)$  it follows that  $f_Y(x^*) = f_Y(x) = 0$ . It follows that  $\mathcal{X}_p = \mathcal{X}_p^Y$ .
3. Let  $\Phi_{<} = (f, <_{<})$  and for any  $x^1, x^2 \in X$  let  $f(x^1) <_{<} f(x^2)$  iff  $x^1 \in \mathcal{X}_p$  and  $x^2 \notin \mathcal{X}_p$ . It follows that  $\mathcal{X}_p = \mathcal{X}_p^{<}$ .

□

Theorem 1 is a powerful result, since it implies that any preference can be expressed by modifying only one component of the corresponding MOOP – either the domain, codomain or the order on the codomain. The same preference can be formulated using different problem transformations. This has major implications for the design and execution of optimization algorithms, since the transformation of each component has a different effect from an algorithmic perspective. Intransitive preference orders, for example, prevent the application of fast sorting techniques for ordering a set of points in the objective space [Eic14].<sup>5</sup> Such techniques, however, are utilized in many popular optimization algorithms in evolutionary computation [DPAM02, BNE07, TFD11, NLA<sup>+</sup>08, BSS15]. Sorting techniques that possess a higher computational complexity must be used instead in these algorithms [Eic14]. Intransitive preferences also allow the existence of preference cycles, which prohibit any ranking between the elements that form part of the cycle. Pareto domination can be used as a fallback mechanism in this case to rank the different points [Bra14]. Restricting the feasible set by introducing additional constraints, on the other hand, allows

<sup>5</sup>Tradeoff domination for example, which is introduced in Section 2.3.2, is an intransitive preference order. There exist equivalent domain and codomain predicates (Sections 2.3.1 and 2.3.3), however that prevent the introduction of intransitivity.

### 2.3. Problem Transformations Using Tradeoff-based Preference Models

the application of any optimization technique that possesses a generic method for handling constraint violations.

At the same time, however, the practical application of the results of Theorem 1 is limited. There exists no straightforward procedure of transforming different preference predicates into each other. It is also possible that the algebraic structure of a problem cannot be maintained through different component transformations. For example, an objective space transformation that is obtained by concatenating a smooth objective function with a non-smooth function, generally results in a transformed problem that possesses a non-smooth objective function [SED13]. Therefore, Theorem 1 should rather serve as a motivation to explore different formulations of the same preference. A practical example for obtaining such different formulations to the same preference is given in the next section.

## 2.3. Problem Transformations Using Tradeoff-based Preference Models

*Declaration: Parts of this section have been published in [BSS11, SBS13, BS16]*

Tradeoffs are a popular and reliable method for describing and eliciting preferences from a DM in MCDA [WB93]. Utilizing tradeoffs requires objectives to be measured on a cardinal preference scale. We therefore consider the real-valued MOOP as of Definition 9 for the remainder of this section. There exist multiple approaches to how tradeoffs can be incorporated in MCDA methodology to identify preferred alternatives. The approach presented in this section is founded on Geoffrion's notion of proper Pareto optimality [Geo68] (see Definition 13 and Figure 2.3).

**Definition 13** (Geoffrion proper Pareto optimality [Geo68]). *An element  $\mathbf{x}^* \in X$  is called Geoffrion proper Pareto optimal if  $\mathbf{x}^* \in X_p$  and if there exists an  $M \in \mathbb{R}_+$  such that for all  $\mathbf{x} \in X$  and  $i \in [m]$  satisfying  $f_i(\mathbf{x}) < f_i(\mathbf{x}^*)$  there exists a  $j \in [m]$  for which  $f_j(\mathbf{x}) > f_j(\mathbf{x}^*)$  and additionally*

$$\frac{f_i(\mathbf{x}^*) - f_i(\mathbf{x})}{f_j(\mathbf{x}) - f_j(\mathbf{x}^*)} \leq M. \quad (2.2)$$

Definition 13 states that a Geoffrion proper Pareto optimal solution must be Pareto optimal and that any deterioration in one objective with respect to another element of  $\mathcal{Y}$  must be bounded by a finite number  $M$ . Geoffrion's concept eliminates Pareto optimal solutions that exhibit an unbounded tradeoff and is thereby a preference as of Definition 11. The notion of Geoffrion proper Pareto

## 2. Preferences in Multi-objective Optimization

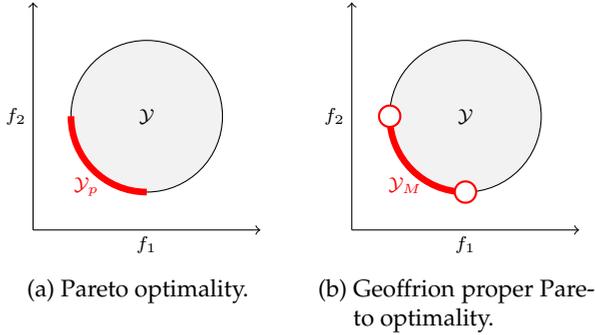


Figure 2.3.: Comparison of Pareto optimality and Geoffrion proper Pareto optimality. The set of Geoffrion proper Pareto optimal elements  $\mathcal{Y}_M$  does not include the boundary points of the Pareto front, since these elements exhibit an unbounded tradeoff as of Equation (2.2).

optimality can be extended by fixing  $M$  to a specific value. This concept is called  $M$ -proper Pareto optimality in [BSS11]. We refer to it as *tradeoff optimality* in this work for brevity, and because this name is a clearer description of the underlying preference notion.

**Definition 14** (tradeoff optimality cf. [Shu07, BSS11]). *Let an  $M \in \mathbb{R}_+$  be given. An element  $\mathbf{x}^* \in X$  is called tradeoff optimal with respect to  $M$  if  $\mathbf{x}^* \in X_p$  and if for all  $\mathbf{x} \in X$  and for all  $i \in [m]$  satisfying  $f_i(\mathbf{x}) < f_i(\mathbf{x}^*)$  there exists a  $j \in [m]$  for which  $f_j(\mathbf{x}) > f_j(\mathbf{x}^*)$  and additionally*

$$\frac{f_i(\mathbf{x}^*) - f_i(\mathbf{x})}{f_j(\mathbf{x}) - f_j(\mathbf{x}^*)} \leq M. \quad (2.3)$$

Definition 14 only differs from Equation (2.2) in  $M$  possessing a fixed value that can be set by a DM.  $M = 5$ , for example, implies that a loss of five units in one objective must be compensated by a gain of at least one unit in another objective. Any tradeoff optimal solution must fulfill this requirement with respect to all feasible elements of the objective space.

### 2.3.1. Tradeoff Domain Predicate

Tradeoff optimality restricts the set of Pareto optimal solutions and, thereby, defines a preference as of Definition 11 that can be expressed by different predicates. In the following, three different approaches are discussed for defining

### 2.3. Problem Transformations Using Tradeoff-based Preference Models

tradeoff optimality predicates that each focus on modifying a single component of the underlying MOOP – domain, codomain and order. This exercise serves as an illustration of the implications of Theorem 1. It also sheds some light on the discussion on how different predicates influence the applicability of subsequent optimization methodologies.

We commence by assessing modifications to the search space. Equation (2.2) can be directly imposed as a constraint on the search space to restrict the set of feasible solutions. By bounding the tradeoff of all feasible elements – not only vectors that map to Pareto optimal solutions – we obtain a constrained problem, whose set of Pareto optimal solutions is the set of tradeoff optimal solutions. The corresponding preference is given in Definition 15.

**Definition 15** (tradeoff domain predicate). *Let  $f = (X, Y, F)$  be the objective function of an MOOP and  $M \in \mathbb{R}_+$ . The tradeoff domain predicate is given by  $(f_X^M, <_p)$  with  $f_X^M = (X_M, Y, F_X^M)$ , where*

- $X_M := \{\mathbf{x} \in X \mid \forall \hat{\mathbf{x}} \in X \text{ and } i \in [m] : f_i(\hat{\mathbf{x}}) < f_i(\mathbf{x}) : \exists j \in [m] \text{ with } f_j(\hat{\mathbf{x}}) > f_j(\mathbf{x}) \text{ and } (f_i(\mathbf{x}) - f_i(\hat{\mathbf{x}}))/(f_j(\hat{\mathbf{x}}) - f_j(\mathbf{x})) \leq M\}$  and
- $F_X^M := \{(\mathbf{x}, \mathbf{y}) \in F \mid \mathbf{x} \in X_M\}$ .

The constraints imposed by  $(f_X^M, <_p)$  on  $X$  are difficult to check during the execution of an optimization algorithm, since the feasibility of an  $\mathbf{x} \in X$  does not solely depend on the value of  $\mathbf{x}$ , but also on other elements of  $X$ , which may be unknown. Although it can be shown in [SHS10a] that it is sufficient to consider only the Pareto optimal set to check a solution for tradeoff optimality, that simplification still requires the complete knowledge of the Pareto front. The Pareto front, however, is usually not known prior to any optimization effort.

The composition of  $X_M$  makes it difficult to solve an MOOP on which a tradeoff domain predicate is imposed if mathematical programming techniques shall be applied. These techniques usually expect feasibility to be expressed by equality and inequality relations that only depend on the decision variables of the element whose feasibility is checked [Ber99]. Even heuristic approaches should be individually adapted. Braun et al. [BSS11] have proposed modifications to four established multi-objective metaheuristics for generating tradeoff optimal solutions. Each modification was tailored to the individual design of every algorithm. This result motivates the exploration of different preference predicates for describing tradeoff optimality. We therefore continue our discussion by assessing order transformations.

## 2. Preferences in Multi-objective Optimization

### 2.3.2. Tradeoff Order Predicate

Equation (2.2) delivers a blueprint for a binary relation that imposes an order for identifying tradeoff optimal solutions. A Pareto optimal solution  $\mathbf{x}^*$  is not tradeoff optimal if there exists a single element  $\mathbf{x}$  such that the fraction in (2.2) becomes larger than the threshold  $M$ . Said fraction is the ratio of the worst deterioration divided by the largest improvement if choosing  $\mathbf{x}^*$  instead of  $\mathbf{x}$ . This tradeoff ratio can be used to define a dominance relation. We can say that  $\mathbf{x}$  dominates  $\mathbf{x}^*$  if the tradeoff that incurs by moving from  $f(\mathbf{x}^*)$  to  $f(\mathbf{x})$  exceeds  $M$ . The concept of tradeoff domination is formalized in Definition 16. Note that tradeoff domination is defined for general vector spaces, since its existence is independent of the optimization context.

**Definition 16** (tradeoff domination [BS16]). *Let  $\mathbf{u}, \mathbf{v} \in \mathbb{R}^m$  and  $M \in \mathbb{R}_+$ . The index sets  $I_{>}$  and  $I_{<}$  are defined by*

$$I_{>} := \{i \in [m] \mid u_i > v_i\}, \quad (2.4a)$$

$$I_{<} := \{i \in [m] \mid u_i < v_i\}. \quad (2.4b)$$

*Then,  $\mathbf{u}$  tradeoff-dominates  $\mathbf{v}$ , denoted by  $\mathbf{u} \prec_t^M \mathbf{v}$ , if either  $\mathbf{u}$  Pareto-dominates  $\mathbf{v}$  or if  $\mathbf{u}$  and  $\mathbf{v}$  are non-dominated and*

$$\max_{i \in I_{<}(\mathbf{u}, \mathbf{v})} \min_{j \in I_{>}(\mathbf{u}, \mathbf{v})} \frac{v_i - u_i}{u_j - v_j} > M. \quad (2.5)$$

Tradeoff domination is called M-domination in [BS16]. The name tradeoff domination was chosen to comply with the naming scheme used in this section. Tradeoff domination is a generalization of U-domination [SBS13], since both notions coincide for  $M = 1$ . Definition 16 can be simplified to eliminate the mandatory Pareto domination check as shown in Proposition 1. The tradeoff order predicate is formally given in Definition 17.

**Proposition 1** ([BS16]). *Let  $\mathbf{u}, \mathbf{v} \in \mathbb{R}^m$  and  $M \in \mathbb{R}_+$ . The expression  $\mathbf{u} \prec_t^M \mathbf{v}$  is equivalent to*

$$\max_{i \in [m]} (v_i - u_i) + M \cdot \min_{i \in [m]} (v_i - u_i) > 0. \quad (2.6)$$

*Proof.* If  $\mathbf{u} \prec_p \mathbf{v}$  then the left side of (2.6) is positive and consequently  $\mathbf{u} \prec_t^M \mathbf{v}$ . If  $\mathbf{v} \prec_p \mathbf{u}$  then the left side of (2.6) is negative and  $\mathbf{u} \not\prec_t^M \mathbf{v}$ . The proof is concluded by showing that (2.5) is equivalent to (2.6) if  $\mathbf{u}$  and  $\mathbf{v}$  are non-dominated:

$$\max_{i \in I_{<}(\mathbf{u}, \mathbf{v})} \min_{j \in I_{>}(\mathbf{u}, \mathbf{v})} \frac{v_i - u_i}{u_j - v_j} > M \quad (2.7a)$$

### 2.3. Problem Transformations Using Tradeoff-based Preference Models

$$\Leftrightarrow \frac{\max_{i \in [m]}(v_i - u_i)}{\max_{i \in [m]}(u_i - v_i)} > M \quad (2.7b)$$

$$\Leftrightarrow \max_{i \in [m]}(v_i - u_i) - M \cdot \max_{i \in [m]}(u_i - v_i) > 0 \quad (2.7c)$$

$$\Leftrightarrow \max_{i \in [m]}(v_i - u_i) + M \cdot \min_{i \in [m]}(v_i - u_i) > 0. \quad (2.7d)$$

□

**Definition 17** (tradeoff order predicate). *Let  $f = (X, Y, F)$  be the objective function of an MOOP and  $M \in \mathbb{R}_+$ . The tradeoff order predicate is given by  $(f, \prec_t^M)$ .*

The tradeoff order predicate (see Definition 17) is usually easier to handle in optimization algorithms compared to the tradeoff domain predicate. Optimization techniques that rely on Pareto domination checks (e.g. [DPAM02, NDGN<sup>+</sup>09, BNE07, BSS15]) can simply replace any occurrence of Pareto domination by tradeoff domination. No additional custom modifications to the algorithm are required. There exist exceptions, however, in case the algorithm exploits mathematical properties of Pareto domination that tradeoff domination does not exhibit. Pareto domination, for example, is a transitive binary relation, whereas tradeoff domination is not as shown in Proposition 2.

**Proposition 2** ([BS16]). *Tradeoff domination is not transitive for  $m \geq 3$  and  $M \in (0, 2)$ .*

*Proof.* Consider the vectors  $\mathbf{u} = (1, -1, 0, \dots, 0)$ ,  $\mathbf{v} = (0, 1, -1, \dots, -1)$  and  $\mathbf{w} = (-1, 0, 1, \dots, 1)$ . It follows that  $\mathbf{u} \prec_t^{(0,2)} \mathbf{v}$ ,  $\mathbf{v} \prec_t^{(0,2)} \mathbf{w}$  and  $\mathbf{u} \not\prec_t^{(0,2)} \mathbf{w}$ . Equivalent examples can be constructed in any underlying vector space. □

The transitivity of Pareto domination is, for example, exploited in nondominated sorting (see Algorithm 2 and Algorithm 2)– a method that is employed by many MOEAs (e.g. [DPAM02, BNE07, DJ14]). Given a set of points  $S \subset \mathcal{Y}$ , nondominated sorting divides  $S$  into subsets of points that do not dominate each other (see Figure 2.4). In case a dominance relation is not transitive, the set of non-dominated elements might be empty. In the proof of Proposition 2, for example,  $\mathbf{w}$  tradeoff dominates  $\mathbf{u}$  resulting in a cyclic ordering of  $\{\mathbf{u}, \mathbf{v}, \mathbf{w}\}$ , in which no minimal elements exist. Imposing the Pareto order on any finite subset of a real-valued vector space, on the other hand, always yields at least one minimum element. In case no minimal elements can be identified during a nondominated sorting step, the corresponding algorithm breaks and can no longer continue its execution. This problem can be solved by employing a fallback to the Pareto order in case no minimal elements exist. The objective space

## 2. Preferences in Multi-objective Optimization

transformation of the tradeoff codomain predicate that is discussed in the final part of this section, provides an opportunity to circumvent this issue.

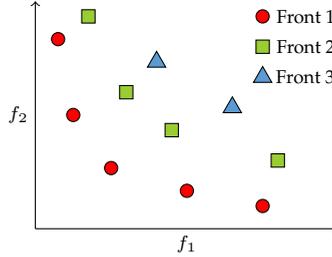


Figure 2.4.: Illustration of nondominated sorting. Elements of a single front do not dominate each other. Each element of front  $i$  is dominated by at least one element of front  $i - 1$ .

### 2.3.3. Tradeoff Codomain Predicate

The final transformation presented in this section is a cone-based decomposition of tradeoff optimality [BS16]. Cones are a concept in algebra that is used in MCDA and MOO to describe dominance relations. A cone constitutes a subset of a vector space with the property that any member of the cone multiplied by a positive constant is again a member of the cone (see Definition 18). Said characteristic is denoted by *cone property*.

**Definition 18** (cone [Wie07]). *A non-empty subset  $C \subset \mathbb{R}^m$  is called cone, if for all  $\mathbf{d} \in C$  and  $\lambda > 0$  it holds that  $\lambda \mathbf{d} \in C$ .*

In MCDA and MOO, any vector  $\mathbf{v}$  that lies in the domination cone of a given point  $\mathbf{u}$ , is considered inferior to  $\mathbf{u}$  (see Figure 2.5). The vector  $\mathbf{d} = \mathbf{v} - \mathbf{u}$  can be interpreted as *preferred direction*. The cone property states that any positive scaling of the difference  $\mathbf{d}$  results in a preferred direction as well [Wie07].

**Definition 19** (cone dominance [BS16]). *Let  $\mathbf{u}, \mathbf{v} \in \mathbb{R}^m$  and  $C \subset \mathbb{R}^m$  be a cone. Vector  $\mathbf{u}$  cone-dominates  $\mathbf{v}$ , denoted by  $\mathbf{u} <_C \mathbf{v}$ , iff  $\mathbf{v} - \mathbf{u} \in C$ .*

Any dominance relation can be used to divide the vector space  $\mathbb{R}^m$  with respect to a  $\mathbf{u} \in \mathbb{R}^m$  into points that are dominated by  $\mathbf{u}$ , points that dominate  $\mathbf{u}$  and points that are non-dominated to  $\mathbf{u}$  (see Definition 20). An illustration of these divisions for tradeoff domination is given in Figure 2.6. Cone-based dominance relations possess the advantage that the cone-dominated space can be directly

### 2.3. Problem Transformations Using Tradeoff-based Preference Models

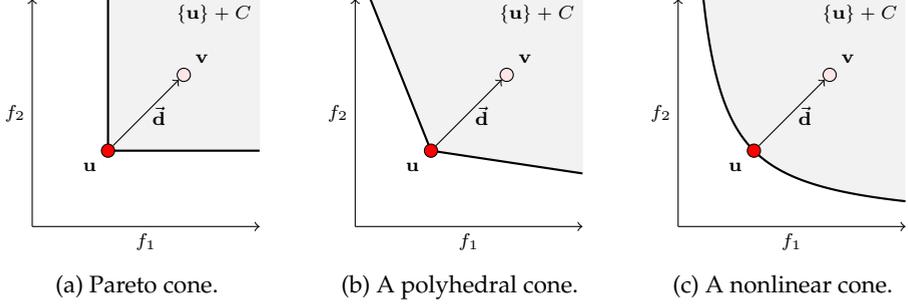


Figure 2.5.: Examples of three different cone types at a point  $\mathbf{u} \in \mathbb{R}^2$ . Any point that lies in the set  $\{\mathbf{u}\} + C$  is cone dominated by  $\mathbf{u}$ . Vector  $\vec{d}$  indicates the preferred direction from  $\mathbf{u}$  to  $\mathbf{v}$ . The cone property (see Definition 18) states that any point  $\mathbf{u} + \lambda \vec{d}$  with  $\lambda > 0$  is cone dominated by  $\mathbf{u}$ .

inferred from the cone definition. Given a cone  $C \in \mathbb{R}^m$ , we obtain  $D_{<C}(\mathbf{u}) = \{\mathbf{u}\} + C$ .

**Definition 20** (dominance, preference and non-dominated space c.f. [BS16]). *Let  $<$  be a binary relation on  $\mathbb{R}^m$ . Points in  $\mathbb{R}^m$  are characterized with respect to a  $\mathbf{u} \in \mathbb{R}^m$  as dominated space  $D_{<}(\mathbf{u})$ , preference space  $P_{<}(\mathbf{u})$  and non-dominated space  $N_{<}(\mathbf{u})$  with*

$$D_{<}(\mathbf{u}) := \{\mathbf{v} \in \mathbb{R}^m \mid \mathbf{u} < \mathbf{v}\}, \quad (2.8a)$$

$$P_{<}(\mathbf{u}) := \{\mathbf{v} \in \mathbb{R}^m \mid \mathbf{v} < \mathbf{u}\}, \quad (2.8b)$$

$$N_{<}(\mathbf{u}) := \mathbb{R}^m \setminus \{D_{<}(\mathbf{u}) \cup P_{<}(\mathbf{u})\}. \quad (2.8c)$$

Cones can be constructed in multiple ways. The focus lies on polyhedral cones (see Definition 22) in this work, since the tradeoff dominated space can be decomposed into polyhedral cones. The common terminology for comparing elements of vector spaces as described in Definition 21 is adopted.

**Definition 21** (vector comparison [Wie07]). *Let  $\mathbf{u}, \mathbf{v} \in \mathbb{R}^m$ . Then,*

$$\mathbf{u} > \mathbf{v} \iff \forall i \in [m] \ u_i > v_i, \quad (2.9a)$$

$$\mathbf{u} \geq \mathbf{v} \iff \forall i \in [m] \ u_i \geq v_i, \quad (2.9b)$$

$$\mathbf{u} \geq \mathbf{v} \iff \mathbf{u} \geq \mathbf{v} \text{ but } \mathbf{u} \neq \mathbf{v}. \quad (2.9c)$$

Polyhedral cones can be used to produce a linear transformation  $Y_C$  of the objective space  $Y$  by multiplying every element of  $Y$  by the matrix  $A$  that induces the polyhedral cone. Thereby, finding the cone-non-dominated points of  $Y$  is

## 2. Preferences in Multi-objective Optimization

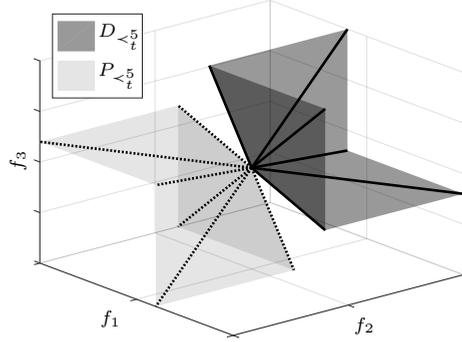


Figure 2.6.: Illustration of the tradeoff dominance and preference space for  $M = 5$  and some  $\mathbf{u} \in \mathbb{R}^3$ .

the same as obtaining the Pareto optimal points of  $Y_C$ . This is a well-known result in MOO [Wie07]. We can exploit this circumstance to formulate a predicate consisting of a codomain transformation to identify tradeoff optimal solutions.

**Definition 22** (polyhedral cone [Wie07]). *Let  $\mathbf{A} \in \mathbb{R}^{l \times m}$  be a matrix with  $l \in \mathbb{N}_+$ . The polyhedral cone  $C(\mathbf{A}) \subset \mathbb{R}^m$  is given by*

$$C(\mathbf{A}) := \{\mathbf{d} \in \mathbb{R}^m \mid \mathbf{A}\mathbf{d} \geq \mathbf{0}\}. \quad (2.10)$$

A cone-based decomposition of the tradeoff dominated space consists of finding a set of polyhedral cones, whose union is equivalent to the tradeoff dominated space (see Figure 2.7). Braun et al. [BS16] have proposed four different decompositions that differ in the number of cones, matrix size and whether individual cones are overlapping (see Table 2.4). Each one of these decompositions is subsequently introduced and discussed.

The ordered objectives approach is based on the idea of dividing the vector space  $\mathbb{R}^m$  into subsets by considering all feasible cases of ordering the elements of the vector  $\mathbf{d} = \mathbf{u} - \mathbf{v}$  from largest to smallest. For example, if  $m = 3$  we have  $\mathbf{d} = (d_1, d_2, d_3)$  and there exist six possibilities to order the elements of  $\mathbf{d}$  (see Table 2.1). In the general case, we obtain  $m!$  different orders for  $m$  objectives. Next, for any such subset, the set of tradeoff dominated points is determined. Since the largest and smallest elements are implicitly known in each division, Equation (2.6) can be expressed as a simple linear inequality without the maximum and minimum function. In turn, the different orderings of the elements in  $\mathbf{d}$  can themselves be expressed by  $m$  inequalities (see Table 2.1).

### 2.3. Problem Transformations Using Tradeoff-based Preference Models

The M-dominated space can thereby be composed of  $m!$  polyhedral cones each induced by an  $m \times m$  matrix (see Theorem 2 and Table 2.2).

Table 2.1.: Linear inequalities of the ordered objectives approach for three objectives.

$d_1 \geq d_2 \geq d_3$	$d_1 \geq d_3 \geq d_2$	$d_2 \geq d_1 \geq d_3$
$d_1 + Md_3 > 0$	$d_1 + Md_2 > 0$	$d_2 + Md_3 > 0$
$d_1 \geq d_2$	$d_1 \geq d_3$	$d_2 \geq d_1$
$d_2 \geq d_3$	$d_3 \geq d_2$	$d_1 \geq d_3$
$d_2 \geq d_3 \geq d_1$	$d_3 \geq d_1 \geq d_2$	$d_3 \geq d_2 \geq d_1$
$d_2 + Md_1 > 0$	$d_3 + Md_2 > 0$	$d_3 + Md_1 > 0$
$d_2 \geq d_3$	$d_3 \geq d_1$	$d_3 \geq d_2$
$d_3 \geq d_1$	$d_1 \geq d_2$	$d_2 \geq d_1$

Table 2.2.: Matrices of the polyhedral cones of the ordered objectives approach for three objectives.

$d_1 \geq d_2 \geq d_3$	$d_1 \geq d_3 \geq d_2$	$d_2 \geq d_1 \geq d_3$
$\begin{pmatrix} 1 & 0 & M \\ 1 & -1 & 0 \\ 0 & 1 & -1 \end{pmatrix}$	$\begin{pmatrix} 1 & M & 0 \\ 1 & 0 & -1 \\ 0 & -1 & 1 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 & M \\ -1 & 1 & 0 \\ 1 & 0 & -1 \end{pmatrix}$
$d_2 \geq d_3 \geq d_1$	$d_3 \geq d_1 \geq d_2$	$d_3 \geq d_2 \geq d_1$
$\begin{pmatrix} M & 1 & 0 \\ 0 & 1 & -1 \\ -1 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 0 & M & 1 \\ -1 & 0 & 1 \\ 1 & -1 & 0 \end{pmatrix}$	$\begin{pmatrix} M & 0 & 1 \\ 0 & -1 & 1 \\ -1 & 1 & 0 \end{pmatrix}$

**Theorem 2** (ordered objectives decomposition [BS16]). *Let  $\mathbf{u} \in \mathbb{R}^m$ ,  $M \in \mathbb{R}_+$  and let  $\Omega$  denote the set of all permutations over the index set  $[m]$ . Furthermore,  $\sigma(i)$  identifies the element at the  $i$ -th position of the permutation vector  $\sigma \in \Omega$ . Then,*

$$D_{<^M}(\mathbf{u}) = \{\mathbf{u}\} + \text{int} \left( \bigcup_{\sigma \in \Omega} C(\mathbf{A}_\sigma^o) \right), \quad (2.11)$$

with

$$\mathbf{A}_\sigma^o = \begin{pmatrix} \mathbf{b}_\sigma^o \\ \mathbf{B}_\sigma^o \end{pmatrix}, \quad (2.12)$$

## 2. Preferences in Multi-objective Optimization

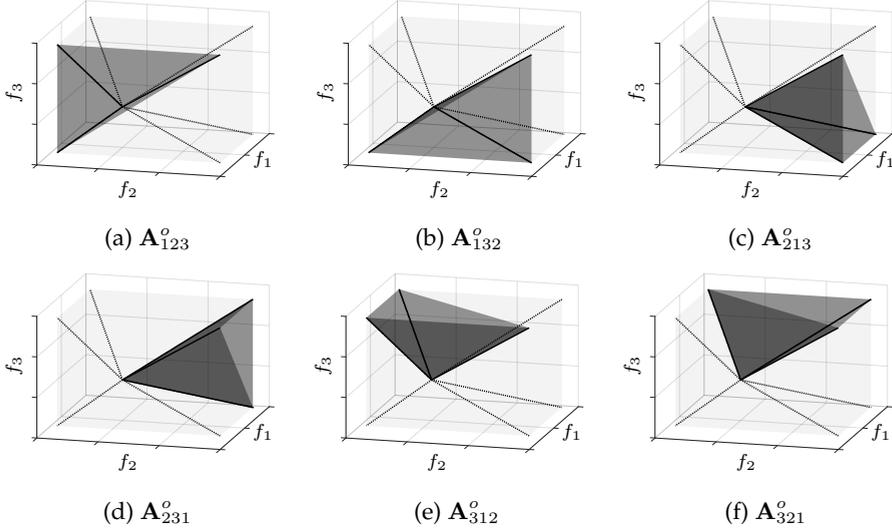


Figure 2.7.: Decomposition of the tradeoff dominated space using the ordered objectives approach for  $M = 3$  and  $m = 3$ . Dark gray patches show the faces of the individual cones of the ordered objectives approach and thick black lines their generators. Light gray patches show the faces of the entire tradeoff dominated space. The dotted lines are the generators of the cone that represents the tradeoff dominated space.

where  $\mathbf{b}_\sigma^o$  is a vector of length  $m$  with

$$\mathbf{b}_\sigma^o(i) := \begin{cases} 1 & \text{if } i = \sigma(1) \\ M & \text{if } i = \sigma(m) \\ 0 & \text{else,} \end{cases} \quad (2.13)$$

and  $\mathbf{B}_\sigma^o$  is an  $(m - 1) \times m$  matrix with

$$\mathbf{B}_\sigma^o(i, j) := \begin{cases} 1 & \text{if } j = \sigma(i) \\ -1 & \text{if } j = \sigma(i + 1) \\ 0 & \text{else.} \end{cases} \quad (2.14)$$

*Proof.* It holds that  $\{\mathbf{u}\} + \bigcup_{\sigma \in \Omega} C(\mathbf{B}_\sigma^o) = \mathbb{R}^m$ . By additionally enforcing the linear inequality given by  $\mathbf{b}_\sigma^o$ ,  $\{\mathbf{u}\} + C(\mathbf{A}_\sigma^o)$  yields exactly those elements in  $C(\mathbf{B}_\sigma^o)$  that are tradeoff dominated by  $\mathbf{u}$  if we relax the strict inequality in Equation (2.6) to a non-strict inequality. Consequently, the union  $\{\mathbf{u}\} + \bigcup_{\sigma \in \Omega} C(\mathbf{A}_\sigma^o)$  yields all elements in  $\mathbb{R}^m$  that are tradeoff dominated by  $\mathbf{u}$  if Equation (2.6) is non-strict. The interior  $\{\mathbf{u}\} + \text{int}(\bigcup_{\sigma \in \Omega} C(\mathbf{A}_\sigma^o))$  is then comprised of all tradeoff dominated points if Equation (2.6) is strict.  $\square$

### 2.3. Problem Transformations Using Tradeoff-based Preference Models

The number of polyhedral cones of the ordered objectives decomposition grows factorially in the number of objectives rendering its application in higher dimensions difficult. This observation yields a motivation for decreasing the number of polyhedral cones that comprise the tradeoff dominated space. Establishing a tradeoff domination relation between two vectors requires only the knowledge their largest and smallest objective difference. The exact order of elements of  $\mathbf{d}$  is irrelevant. If  $d_i$  is the largest and  $d_j$  the smallest component of  $\mathbf{d}$ , it is sufficient to require that all other components are smaller than  $d_i$  and larger than  $d_j$ . This approach is referred to as min/max decomposition (Definition 23). The number of matrices is obtained as the number of all 2-permutations of  $m$  yielding  $m(m-1)$  matrices in total.

**Definition 23** (min/max decomposition [BS16]). *Let  $\mathbf{u} \in \mathbb{R}^m$ ,  $M \in \mathbb{R}_+$  and  $\mathbf{s}^{ij} = (c)_{c=1, c \neq i, c \neq j}^m$  with  $i, j \in [m]$ , i.e. the tuple of natural numbers from 1 to  $m$  without  $i$  and  $j$ .*

$$D_{<_t^M}(\mathbf{u}) = \{\mathbf{u}\} + \text{int} \left( \bigcup_{i \in [m]} \bigcup_{j \in [m] \setminus \{i\}} C(\mathbf{A}_{ij}^{mm}) \right), \quad (2.15)$$

where

$$\mathbf{A}_{ij}^{mm} = \begin{pmatrix} \mathbf{b}_{ij}^M \\ \mathbf{B}_{ij}^{mm} \\ -\mathbf{B}_{ji}^{mm} \end{pmatrix}, \quad (2.16)$$

with

$$\mathbf{b}_{ij}^M(k) := \begin{cases} 1 & \text{if } k = i \\ M & \text{if } k = j \\ 0 & \text{else,} \end{cases} \quad (2.17)$$

and

$$\mathbf{B}_{ij}^{mm}(k, l) := \begin{cases} 1 & \text{if } k = i \\ -1 & \text{if } l = s_k^{ij} \\ 0 & \text{else.} \end{cases} \quad (2.18)$$

*Proof.* It holds that  $\{\mathbf{u}\} + \bigcup_{i \in [m]} \bigcup_{j \in [m] \setminus \{i\}} C([\mathbf{B}^{mm} - \mathbf{B}_{ij}^{mm}]^T) = \mathbb{R}^m$ . Cone  $\{\mathbf{u}\} + C(\mathbf{A}_{ij}^{mm})$  then contains the closure of the elements in  $\{\mathbf{u}\} + C([\mathbf{B}^{mm} - \mathbf{B}_{ij}^{mm}]^T)$  that are tradeoff dominated by  $\mathbf{u}$ . Their union forms the closure of the space that is tradeoff-dominated by  $\mathbf{u}$ .  $\square$

For three objectives, the ordered objectives and the min/max decomposition produce the same polyhedral cones. Both approaches differ in four and higher dimensions. Therefore, an illustrative example of the min/max decomposition for  $m = 5$  is provided in Example 1.

## 2. Preferences in Multi-objective Optimization

**Example 1.** The matrix  $\mathbf{A}_{ij}^{mm}$  for  $m = 5$ ,  $M = 3$ ,  $i = 1$  and  $j = 2$ :

$$\mathbf{A}_{12}^{mm} = \begin{pmatrix} 1 & 3 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & 0 & -1 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 & 1 \end{pmatrix}. \quad (2.19)$$

The number of polyhedral cones can be further reduced by decomposing Equation (2.5) into  $m - 1$  inequalities. Let  $d_m$  be the largest component of  $\mathbf{d}$ . The condition stated in Equation (2.5) can be equivalently formulated as  $d_m + Md_i > 0$  for all  $i \in [m - 1]$  resulting in  $m - 1$  inequalities. The component  $d_m$  can be established as largest element of  $\mathbf{d}$  by the inequalities  $d_m > d_i$  for all  $i \in [m - 1]$ . The resulting  $2m - 2$  inequalities describe the tradeoff dominated space if  $d_1$  is the largest component of  $\mathbf{d}$ . The number of polyhedral cones is thereby reduced to  $m$  (Table 2.3 and Figure 2.8). This approach is called maximum decomposition, since only the maximum component of  $\mathbf{d}$  must be known (Definition 24).

Table 2.3.: Matrices of the polyhedral cones of the maximum approach for three objectives.

$d_1 = \max_{i \in [3]}(d_i)$	$d_2 = \max_{i \in [3]}(d_i)$	$d_3 = \max_{i \in [3]}(d_i)$
$\begin{pmatrix} 1 & -1 & 0 \\ 1 & 0 & -1 \\ 1 & M & 0 \\ 1 & 0 & M \end{pmatrix}$	$\begin{pmatrix} -1 & 1 & 0 \\ 0 & 1 & -1 \\ M & 1 & 0 \\ 0 & 1 & M \end{pmatrix}$	$\begin{pmatrix} -1 & 0 & 1 \\ 0 & -1 & 1 \\ 3 & 0 & 1 \\ 0 & 3 & 1 \end{pmatrix}$

**Definition 24** (maximum decomposition [BS16]). Let  $\mathbf{u} \in \mathbb{R}^m$ ,  $M \in \mathbb{R}_+$  and  $\mathbf{s}^i = (c)_{c=1, c \neq i}^m$  with  $i \in [m]$ , i.e. the tuple of natural numbers from 1 to  $m$  without  $i$ .

$$D_{<^M}(\mathbf{u}) = \{\mathbf{u}\} + \text{int} \left( \bigcup_{i \in [m]} C(\mathbf{A}_i^{\max}) \right), \quad (2.20)$$

with

$$\mathbf{A}_i^{\max} = \begin{pmatrix} \mathbf{B}_i^M \\ \mathbf{B}_i^{\max} \end{pmatrix}, \quad (2.21)$$

### 2.3. Problem Transformations Using Tradeoff-based Preference Models

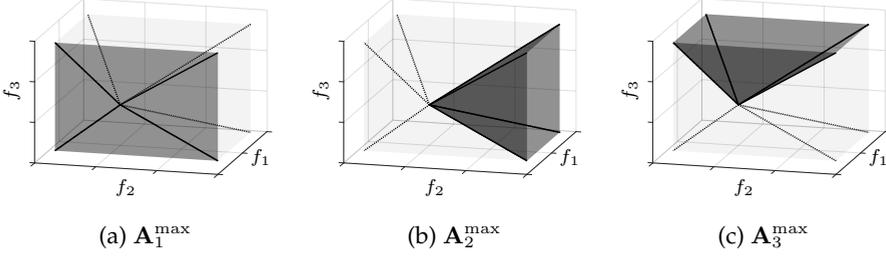


Figure 2.8.: Decomposition of the tradeoff dominated space using the maximum approach for  $M = 3$  and  $m = 3$ . Dark gray patches show the faces of the individual cones of the ordered objectives approach and thick black lines their generators. Light gray patches show the faces of the entire tradeoff dominated space. The dotted lines are the generators of the cone that represents the tradeoff dominated space.

where

$$\mathbf{B}_i^M(k, l) := \begin{cases} 1 & \text{if } k = i \\ M & \text{if } l = s_k^i \\ 0 & \text{else,} \end{cases} \quad (2.22)$$

and

$$\mathbf{B}_i^{\max}(k, l) := \begin{cases} 1 & \text{if } k = i \\ -1 & \text{if } l = s_k^i \\ 0 & \text{else.} \end{cases} \quad (2.23)$$

*Proof.* It holds that  $\{\mathbf{u}\} + \bigcup_{i \in [m]} C(\mathbf{B}_i^{\max}) = \mathbb{R}^m$ . Cone  $\{\mathbf{u}\} + C(\mathbf{A}_{ij}^{\max})$  then contains the closure of the elements in  $\{\mathbf{u}\} + C(\mathbf{B}_i^{\max})$  that are tradeoff dominated by  $\mathbf{u}$ . Their union forms the closure of the space that is tradeoff-dominated by  $\mathbf{u}$ .  $\square$

The size of the matrices in the maximum decomposition can be further reduced by substituting the linear inequalities for enforcing that  $d_i$  is the largest component in  $\mathbf{A}_i^{\max}$  by a single inequality that bounds the tradeoff of  $d_i$  to all other components of  $\mathbf{d}$ . The inequalities implied by  $\mathbf{B}_i^{\max}$  establish  $d_i$  as the largest component of  $\mathbf{d}$ . This requirement can be relaxed by stating that for an arbitrary  $j \in [m] \setminus \{i\}$  the relation  $d_j + Md_i > 0$  must hold. This condition allows other components in  $\mathbf{d}$  to be larger than  $d_i$ , however not so large such that Equation (2.5) is violated. By transitivity of the  $>$ -relation it holds that if  $d_i + Md_k > 0$  for all  $k \in [m] \setminus \{j\}$  and  $d_j + Md_i > 0$  for a single  $j \in [m] \setminus \{j\}$  then  $d_k + Md_i > 0$  for all  $k$ . This approach is referred to by minimum decomposition, since it utilizes the minimum number of matrices and matrix rows (Definition 25). Note

## 2. Preferences in Multi-objective Optimization

that  $j$  can be chosen arbitrarily. It is chosen as  $i + 1$  in Definition 25. The relaxation of the minimum decomposition, however, comes at the cost of the cones no longer being disjunctive sets. Instead, the cones overlap resulting in the sum of the volume of the individual cones being larger than the volume of tradeoff dominated space (Figure 2.9). This is problematic in volume-based optimization techniques that usually require an exact decomposition of the dominated space [BNE07].

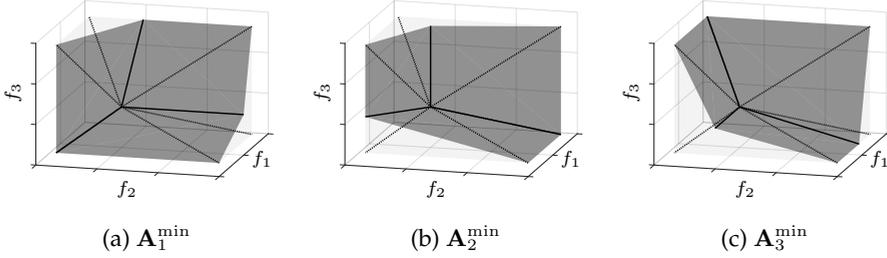


Figure 2.9.: Decomposition of the tradeoff dominated space using the minimum approach for  $M = 3$  and  $m = 3$ . Dark gray patches show the faces of the individual cones of the ordered objectives approach and thick black lines their generators. Light gray patches show the faces of the entire tradeoff dominated space. The dotted lines are the generators of the cone that represents the tradeoff dominated space.

**Definition 25** (minimum decomposition [BS16]). *Let  $\mathbf{u} \in \mathbb{R}^m$  and  $M \in \mathbb{R}_+$ .*

$$D_{<^M}(\mathbf{u}) = \{\mathbf{u}\} + \text{int} \left( \bigcup_{i \in [m]} C(\mathbf{A}_i^{\min}) \right), \quad (2.24)$$

with

$$\mathbf{A}_i^{\min} = \begin{pmatrix} \mathbf{B}_i^M \\ \mathbf{b}_i^M \end{pmatrix}, \quad (2.25)$$

where

$$\mathbf{b}_i^M(k) = \begin{cases} 1 & \text{if } k = i \\ M & \text{if } k = (i \bmod m) + 1 \\ 0 & \text{else.} \end{cases} \quad (2.26)$$

*Proof.* It is sufficient to show that  $C(\mathbf{A}_i^{\max}) \subseteq C(\mathbf{A}_i^{\min})$  and that  $C(\mathbf{A}_i^{\min})$  only contains tradeoff dominated elements. Let  $d_i$  be the largest component of  $\mathbf{d}$ . If  $\mathbf{B}_i^{\max} \mathbf{d} \geq \mathbf{0}$  then  $d_k + M d_i \geq 0$  for all  $k \in [m] \setminus \{i\}$  implying that  $C(\mathbf{A}_i^{\max}) \subseteq C(\mathbf{A}_i^{\min})$ . If  $d_i$  is not the maximum element of  $\mathbf{d}$ , then  $d_k + M d_i \geq 0$  only holds

### 2.3. Problem Transformations Using Tradeoff-based Preference Models

for all  $k$  if  $\arg \max_{j \in [m]}(d_j) + M \arg \min_{j \in [m]}(d_j) \geq 0$ , which is equivalent to  $\mathbf{d} \in D_{<_t^M}(\mathbf{u})$ .  $\square$

Any of the presented cone based decompositions can be used to formulate a tradeoff codomain predicate. The formulation of the tradeoff domain predicate as of Definition 26, however, suffers from two deficiencies. Since the tradeoff dominated space is a composition of multiple polyhedral cones, each element of the transformed codomain comprises a tuple of vectors – one vector for each matrix. The Pareto order cannot be directly imposed on such elements. Instead, individual vectors are compared on a pair-wise basis for Pareto domination. If any vector  $\mathbf{A}_i \mathbf{u}$  Pareto dominates an  $\mathbf{A}_i \mathbf{v}$  with  $\mathbf{u}, \mathbf{v} \in \mathbb{R}^m$ , then,  $\mathbf{u}$  tradeoff dominates  $\mathbf{v}$ . In this sense, Definition 26 is not a strict transformation that changes only the codomain, although the Pareto order builds the basis of comparison. Secondly, by letting Equation (2.6) be non-strict, we remove those elements from the tradeoff optimal set for which equality holds in Equation (2.3) – these are those elements that are located exactly at the tradeoff boundary. Although this circumstance prohibits a mathematical equivalence between the codomain predicate and the domain and order predicate, this limitation is negligible in a practical context, since the preferred sets identified by the different predicates nearly coincide.

**Definition 26** (tradeoff codomain predicate). *Let  $f = (X, Y, F)$  be the objective function of an MOOP and  $M \in \mathbb{R}_+$ . Let  $A := (\mathbf{A}_1, \dots, \mathbf{A}_k)$  be the generator matrices of a cone based decomposition of  $D_{<_t^M}(\mathbf{u})$  with  $\mathbf{u} \in \mathbb{R}^m$ . Furthermore, let the inequality in (2.6) be non-strict. The tradeoff codomain predicate is given by  $(f_Y^M, <_Y)$  with  $f_Y^M = (X, Y_M, F_Y^M)$ , where*

- $Y_M := \{(\mathbf{A}_1 \mathbf{y}, \dots, \mathbf{A}_k \mathbf{y}) \mid \mathbf{y} \in \mathbb{R}^m\}$ ,
- $F_Y^M := \{(\mathbf{x}, (\mathbf{A}_1 \mathbf{y}, \dots, \mathbf{A}_k \mathbf{y})) \mid (\mathbf{x}, \mathbf{y}) \in F\}$  and
- For all  $\mathbf{y}^1, \mathbf{y}^2 \in Y_M$  the relation  $\mathbf{y}^1 <_Y \mathbf{y}^2$  implies there exists an  $i \in [k]$  such that  $\mathbf{A}_i \mathbf{y}^1 <_p \mathbf{A}_i \mathbf{y}^2$ .

A common approach in MOO is to approximate the Pareto front by a finite set of points (see Section 4.1 for a detailed discussion on the topic). Usually, these approximations aim at finding a uniform distribution of points on the Pareto front in the objective space. Since uniform distributions on manifolds in vector spaces can be described by concurrent definitions, there exist multiple approaches to define optimal distributions of points [Deb01, CCLVV07]. Strict codomain predicates possess an interesting feature in this regard that sets them apart from domain and order predicates. Said optimal distribution notions are all defined for the Pareto order, since the entire Pareto front is approximated. A strict codomain predicate transforms the objective space of an MOOP, such

## 2. Preferences in Multi-objective Optimization

that the Pareto order is imposed on the transformed objective space. Therefore, these distribution notions can be directly applied to the transformed objective space and thereby generate an approximation to the Pareto front that is both optimal with respect to said distribution notion and the domain predicate.

In the present context of the tradeoff codomain predicate, the application of optimal distribution concepts is more complicated, since the Pareto order cannot be directly imposed on the tradeoff transformed objective space. Their application, however, is nonetheless possible. The quality of Pareto front approximations is usually measured by indicators [ZT99, BSS15, VVL98]. Such indicator values can be computed for every objective space transformation induced by individual cones and subsequently summed.

Table 2.4 provides an overview and comparison of the decompositions introduced in this section. None of the proposed decompositions is best in each category used for comparison. A closer look at Table 2.4 reveals that the approach using the smallest number of cones still requires  $m$  cones. It is of interest to assess, which is the minimum number of polyhedral cones for decomposing the tradeoff dominated space. In the end, if it was feasible to describe the tradeoff dominated space by a single polyhedral cone  $C$ , it would be possible to formulate a strict tradeoff codomain predicate, since Pareto domination could be directly imposed on the transformed objective space invoked by the generator matrix of  $C$ .

Table 2.4.: Comparison of decomposition approaches of the tradeoff dominated space by polyhedral cones. A decomposition is considered to be overlapping if the cut of any two cones forms a space of  $m$  dimensions [BS16].

	Number of cones	Rows per matrix	Overlapping
Ordered objectives	$m!$	$m$	No
Min/max	$m(m-1)$	$2(m-2)+1$	No
Maximum	$m$	$2(m-1)$	No
Minimum	$m$	$m$	Yes

In determining the minimum number of cones, the mathematical property of convexity for characterizing cones is discussed first (see Definition 27). Convexity is often demanded to guarantee a certain degree of preference consistency. Said property enforces the additivity of preference directions. Any combination of moves in preferred directions must itself result in a preferred direction.

**Definition 27** (convex cone [Wie07]). *A cone  $C$  is called convex iff for all  $\mathbf{d}^1, \mathbf{d}^2 \in C$  it holds that  $\mathbf{d}^1 + \mathbf{d}^2 \in C$ .*

### 2.3. Problem Transformations Using Tradeoff-based Preference Models

The tradeoff dominated space is not convex as shown in Proposition 3. It is a well known result in MOO, however, that polyhedral cones are convex (see [Wie07]). This implies that the tradeoff dominated space cannot be described by a single polyhedral cone. This result can be extended by identifying the subsets of the tradeoff dominated space that render it non-convex. The line connecting any two points in such two different subsets lies partially outside of the tradeoff dominated space. As shown in Corollary 1, there exist  $m$  such subsets and thereby we require at least  $m$  polyhedral cones to construct the tradeoff dominated space.

**Proposition 3.** *Let  $M \in \mathbb{R}_+$  and  $\mathbf{u} \in \mathbb{R}^m$  be arbitrary. The tradeoff dominated space  $D_{<_t^M}(\mathbf{u})$  is not convex if  $m \geq 3$ .*

*Proof.* Without loss of generality, let  $\mathbf{u} = (0, \dots, 0)$  and  $\mathbf{v}^i$  denote the vector of negative ones of length  $m$ , whose  $i$ -th entry is equal to  $M + \varepsilon$  with  $\varepsilon \in (0, M]$ . Let us consider the point  $\mathbf{w} = \mathbf{v}^1/2 + \mathbf{v}^2/2$ . If  $D_{<_t^M}(\mathbf{u})$  was convex, then  $\mathbf{w} \in D_{<_t^M}(\mathbf{u})$  according to Definition 18. However,  $\mathbf{w} = ((M + \varepsilon - 1)/2, (M + \varepsilon - 1)/2, -1, \dots, -1)$  and if  $m \geq 3$  we obtain

$$\max_{i \in [m]} (w_i - u_i) + M \min_{i \in [m]} (w_i - u_i) < 0 \quad (2.27a)$$

$$\Leftrightarrow \frac{1}{2} (M + \varepsilon - 1) + M(-1) < 0 \quad (2.27b)$$

$$\Leftrightarrow \varepsilon - M - 1 < 0, \quad (2.27c)$$

which implies  $\mathbf{u} \not\prec_t^M \mathbf{w}$ . □

**Corollary 1.** *Let  $M \in \mathbb{R}_+$  and  $\mathbf{u} \in \mathbb{R}^m$ . Decomposing the tradeoff dominated space  $D_{<_t^M}(\mathbf{u})$  into polyhedral cones requires at least  $m$  cones if  $m \geq 3$ .*

*Proof.* Without loss of generality, it has been established in the proof of Proposition 3 that for any  $j, k \in [m]$  and  $j \neq k$  it holds that  $\mathbf{w}^{j,k} = \mathbf{v}^j/2 + \mathbf{v}^k/2 \notin D_{<_t^M}(\mathbf{u})$  if  $m \geq 3$ . This implies that no  $\mathbf{v}^j$  and  $\mathbf{v}^k$  can be located in the same polyhedral cone. Since there exist  $m$  vectors  $\mathbf{v}^i$  we conclude that they all must be located in distinct polyhedral cones. □

The chapter is concluded by illustrating the limiting behavior of the tradeoff dominated space for extreme values of  $M$  (Figure 2.10). For  $M \rightarrow 0$ , any point  $\mathbf{u} \in \mathbb{R}^m$  tradeoff dominates all those points that do not lie in its negative orthant. If  $M \rightarrow \infty$  the tradeoff dominated space converges to the Pareto cone. In this sense, tradeoff domination is a generalization of Pareto domination and thereby an extension of a fundamental concept in MOO.

## 2. Preferences in Multi-objective Optimization

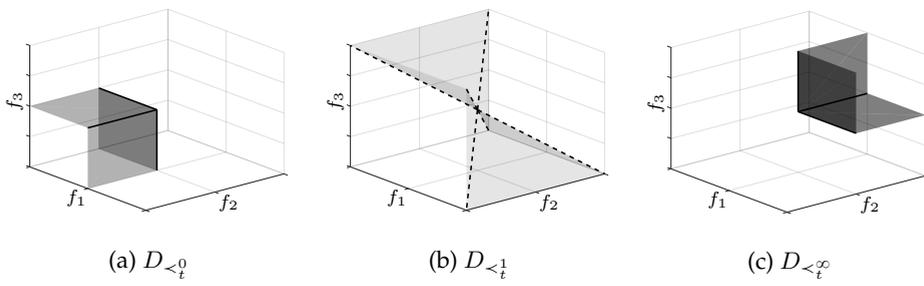


Figure 2.10.: Illustration of the tradeoff dominated space for extreme values.

## 3. Scalarized Preferences

The scope of this chapter is providing a better understanding of the methodology of scalarization by presenting a framework for categorizing and characterizing scalarization functions. A system of axioms is proposed to describe scalarization functions. These axioms represent desirable properties a scalarization function may or should exhibit and thereby provide means to identify the strengths and weaknesses of individual scalarization functions. This system of axioms can thereby help a DM in choosing a scalarization function that is a suitable representation of her preferences. The framework is illustrated by characterizing multiple existing scalarization functions, which are revisited in later chapters. These functions have been explicitly chosen to exemplify specific challenges and difficulties that are frequently encountered in MOO.

The general notion of scalarized preferences is discussed in Section 3.1 and a formal definition of scalarization functions is given. A broad categorization is proposed for classifying scalarization functions that is based on the arithmetic operations involved in computing scalarization values. Multiple instances of scalarization functions that are used in MOO and throughout this work are presented and the relationships between them are explored. Section 3.2 introduces a system of axioms to characterize and describe scalarization functions. Special consideration is given to describing how different scales of measuring objectives affects the ranking of Pareto optimal points induced by scalarization functions. Mathematical proofs are given to show which scalarization functions introduced in Section 3.1 fulfill the proposed axioms. It is shown that all presented scalarization functions satisfy only subsets of the proposed axioms, requiring the DM to carefully weigh advantages and disadvantages in choosing a function that represents her preferences.

### 3.1. The Scalarized Preference Model

*Declaration: Parts of this section have been published in [BSS17] and [SBS13].*

Scalarization is a technique for solving MOOPs, see Section 4.1 for a discussion on solving paradigms in MOO. Said method is used in a two-fold way in MOO: 1) obtaining either the entire set of Pareto optimal solutions in multi-objective

### 3. Scalarized Preferences

combinatorial optimization (MOCO) [GEF16, Chapter 19] or a discretized, uniform subset of the Pareto front for general continuous problems [Mie99]; 2) expressing preferences and identifying a single solution that is preferred to all other options [Deb01, CCLVV07]. Scalarization functions are also known as *value functions* in this context [Mie99, Sec. 4.1]. The analysis in this thesis focuses solely on the latter case.

Scalarized preferences are a well-known technique for solving MOOP, which finds widespread application in research and in practice [MA04]. Scalarization functions are conceptually attractive, since they eliminate the necessity of choice by identifying a single globally optimal solution. They are either derived naturally from the objectives being optimized – for example the efficiency of a combustion process [BSE<sup>+</sup>16] - or are the result of expert preference elicitation [GEF16, KR93]. In the latter case, the application of a scalarization function requires a profound understanding of mathematical preference modeling to fully capture the DM’s preference [Mie99, Sec. 4.1]. The analysis in this section is aimed at providing assistance in the elicitation step. The case in which scalarization functions are used to compute the entire Pareto front or a discretized subset of it is omitted, since they do not involve DM preferences.

The real-valued MOOP as of Definition 9 is considered as basis of the discussion for clarity and simplicity. Generally speaking, a scalarization function is a map from the objective space to the set of real numbers (see Definition 28). We assume that the scalarization function is minimized to comply with the notion of minimizing individual objectives.

**Definition 28** (scalarization function). *A scalarization function is a map  $\Psi = (X, Y, F)$  with  $X \subseteq \mathbb{R}^m$  and  $Y \subseteq \mathbb{R}$ .*

Given an MOOP with objective function  $f$  and a scalarization function  $\Psi$ , a preferred solution is obtained by finding a minimum of the composition  $\Psi \circ f$ . The domain of the scalarization function is thereby restricted to the feasible set of the objective space  $\mathcal{Y}$ . The application of a scalarization function for obtaining a preferred solution can be formulated as predicate as of Definition 12 (see Definition 29). The scalarization predicate consists of an objective space transformation. A scalarized preference reduces the Pareto optimal set to those points that map to the smallest obtainable scalarization value. If the scalarization optimum is unique, the preferred subset (Definition 11) consists of a single element.

**Definition 29** (scalarization predicate). *Let  $f = (X, Y, F)$  be the objective function of an MOOP. A predicate  $(f^s, \leq)$  with  $f^s = (X_s, \mathbb{R}, F_s)$  is called scalarization predicate if  $X_s = X$ .*

- The codomain of  $f^s$  is referred to as scalarization space,

### 3.1. The Scalarized Preference Model

- $\mathbf{x}^s = \arg \min_{\mathbf{x} \in X} f^s(\mathbf{x})$  with  $f(\mathbf{x}^s)$  being Pareto optimal is called scalarization optimizer,
- $f(\mathbf{x}^s)$  is called optimal scalarization value or scalarization optimum.

By mapping vectors of objectives to real numbers, a total order is invoked on the objective space that allows ranking all of its elements. The Pareto optimal solution that ranks at the top of all other alternatives is considered the most preferred choice. The mathematical structure of the scalarization function is decisive to how scalarization values are assigned to vectors of the objective space. Depending on the chosen function, individual rankings may vary greatly [MA04, Mie99]. A DM is therefore required to select a function that is a suitable representation of her preferences. This task, however, may just be as difficult as manually selecting a solution from a given set of alternatives. MCDA methodology can be used to alleviate this process [GEF16].

In the following, three base categories of scalarization functions are proposed. Thereafter, multiple instances of scalarization functions used in research and in practice are presented. These functions have been chosen to reflect the large array of scalarization methods available in MOO (see [MA04, GC17] for an overview). Special consideration was given to the algebraic structure of the different functions by including a range of functions that utilize simpler operations such as addition or multiplication and more complex operations such as norms or trigonometric functions. These operations lead to the functions exhibiting different mathematical properties such as smoothness or jump discontinuities. The presented functions also possess a varying degree of additional information required to correctly compute scalarization values. Such information includes reference points such as the vector of smallest attainable objective values (see Definition 35) that must be precomputed in advance. The algebraic structure and the required additional information both pose different challenges in the applicability of algorithms to obtain preferred solutions. The presented functions include functions that are frequently used by researchers and practitioners such as the weighted sum (Definition 33) and the Chebyshev method (Definition 36), but also methods that have been presented more recently (see Definitions 41 and 42).

#### 3.1.1. Additive Scalarization Function

To facilitate a better understanding of the scalarization methodology, a categorization of scalarization functions is proposed. This categorization is revisited in the next section when different properties of scalarization functions of these categories are discussed. The additive model that has been briefly discussed in

### 3. Scalarized Preferences

Section 1.2 builds the basis of the first category that is considered. The additive scalarization function (see Definition 30) transforms individual objective values and sums them up. These transformations can take place to accommodate different valuations for each objective. Concave utility functions, for example, are used in economics to express the diminishing marginal utility of income and wealth [VNM53]. The transformations may be arbitrary and different for each objective. The only requirement is that they are strictly increasing, so the natural ordering of the objectives is not partially reversed, i.e. a transformation must not lead to larger objective values being preferred to smaller ones.

**Definition 30** (additive scalarization function cf. [GEF16]). *Let  $\{g_1, \dots, g_m\}$  be a set of strictly increasing functions with  $g_i : \mathbb{R} \rightarrow \mathbb{R}$  for all  $i \in [m]$ . The additive scalarization function is given by*

$$\Psi^+(\mathbf{u}) := \sum_{i \in [m]} g_i(u_i). \quad (3.1)$$

#### 3.1.2. Multiplicative Scalarization Function

The multiplicative scalarization function (see Definition 31) has also been briefly discussed in Section 1.2. It is closely related to the additive scalarization function. Instead of summing the transformed objectives, the values are multiplied. Thereby, small increases or decreases in individual objectives may exert larger changes in the scalarization value in comparison to additive functions. Any multiplicative function can be transformed into an additive function by logarithmizing [GEF16]. Although both models – the additive and multiplicative scalarization function – are isomorphic if transformed objective values are positive, their applicability in optimization algorithms greatly differs because of their different algebraic properties. The multiplicative function, for example, usually introduces nonlinearities in the problem formulation that might be difficult to handle using mathematical programming techniques (see Section 4.2 for a discussion on different optimization methodologies). This, in turn, increases the effort for validating necessary and sufficient conditions to check for optimality, since these usually require the computation of derivatives [Ber99, Mie99, MA04]. Logarithmizing a multiplicative function, on the other hand, is also expected to result in nonlinearities that might be even more difficult to handle in comparison to using the product.

**Definition 31** (multiplicative scalarization function cf. [GEF16]). *Let  $\{g_1, \dots, g_m\}$  be a set of strictly increasing functions with  $g_i : \mathbb{R} \rightarrow \mathbb{R}$  for all  $i \in [m]$ . The multiplicative scalarization function is given by*

$$\Psi^*(\mathbf{u}) := \prod_{i \in [m]} g_i(u_i). \quad (3.2)$$

### 3.1. The Scalarized Preference Model

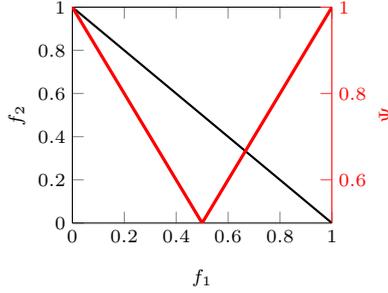


Figure 3.1.: Illustration of the non-differentiability of the maximal scalarization function. The Pareto front is given by  $\mathcal{Y}_p := \{(x, 1-x) \in \mathbb{R}^2 \mid x \in [0, 1]\}$  and the scalarization function as  $\Psi(\mathbf{u}) = \max(u_1, u_2)$ . The function  $\Psi$  is non-differentiable at the point  $(0.5, 0.5)$ , where the maximizing objective of  $\Psi$  changes from  $f_2$  to  $f_1$ .

#### 3.1.3. Maximal Scalarization Function

The third and final category in this thesis is the maximal scalarization function (see Definition 32). This class of functions computes the scalarization value as the maximum of all transformed objective values. In that sense, the objective that is fulfilled worst is decisive for the desirability of a solution. Maximal functions usually introduce non-differentiability at those points, at which maximizing objective of  $\Psi^{\max}$  changes (see Figure 3.1). Non-differentiability prohibits the application of many efficient mathematical programming techniques [Ber99]. This drawback can be remedied, however, by reformulating the maximum function as a constraint [Mie99]. Adding constraints to the problem formulation, on the other hand, usually increases the required effort for solving the problem [MA04].

**Definition 32** (maximal scalarization function cf. [MA04, Mie99]). *Let  $\{g_1, \dots, g_m\}$  be a set of strictly increasing functions with  $g_i : \mathbb{R} \rightarrow \mathbb{R}$  for all  $i \in [m]$ . The maximal scalarization function is given by*

$$\Psi^{\max}(\mathbf{u}) := \max_{i \in [m]} g_i(u_i). \quad (3.3)$$

Many scalarization functions that have been proposed throughout the MOO literature are instances of either the additive, multiplicative or maximal scalarization function [MA04]. These three functions therefore serve as baseline models for categorizing scalarized preferences. Subsequently, any mathematical characterization of the additive, multiplicative and maximal scalarization functions

### 3. Scalarized Preferences

applies to every instance of these baseline models as well. The characterizations of these three functions presented in Section 3.2 therefore apply to a large array of scalarization functions utilized in MOO methodology. Definitions 30 to 32, however, are not exhaustive for all existing and conceivable scalarization functions. Neither one of these three functions addresses interdependencies between objectives, for example. Many scalarization approaches that have been developed in recent years, use complex mathematical computations that cannot be easily subsumed under joint categories [DG11, Das99, SBS13, BSS17]. Nonetheless, these approaches share many similarities with the three baseline models that further the understanding into said approaches. It must be noted that equivalent formulations for Definitions 30 to 32 exist for strictly decreasing transformation functions. In this case, Equations (3.1) to (3.3) must be negated to comply with minimization.

The remainder of this section introduces several explicit instances of scalarization functions that are applied in MOO. These functions serve as a basis for discussing the characterizations in Section 3.2 and are revisited later in the computational studies in Chapters 5 and 6. They were chosen for this thesis, because they use different notions for representing preferences and because they vary in their degree of computational complexity – the necessary effort for computing scalarization values.

The scalarization functions that are introduced in the following can be categorized as either *parameter-dependent* or *parameter-free* (see Table 3.1). Parameter-dependent scalarization functions require the DM to reveal additional preference information used for prioritizing the different objectives. The parameter-dependent scalarization functions presented in this section require the specification of weights to describe the relative importance of each objective. There exist, however, other scalarization techniques that require more complex preference information such as reference points [TMKM09] or desirability ranges for each objective [Mes96]. Parameter-free scalarization functions, on the other hand, do not require any additional information by the DM to compute scalarization values. Scalarization functions that are parameter-free assume that all objectives are equally important. In this sense, they deem a solution preferable if it strikes a balance between all objectives. The scalarization functions that are used in the computational studies in Chapters 5 and 6 are parameter-free to facilitate a comparison across different scalarization functions that is unbiased of any prioritization between different objectives.

#### 3.1.4. Weighted Sum and Sum of Objectives

The weighted sum method (see Definition 33) is one of the most popular scalarization functions in MCDA and MOO. It is an instance of the additive scalar-

### 3.1. The Scalarized Preference Model

Scalarization function	Symbol	Parameter	Reference	Base model
Angle utility	$\Psi^a$	free	Definition 41	None
Chebyshev	$\Psi^c$	free	Definition 36	Maximal
Nash bargaining solution	$\Psi^n$	free	Definition 39	Multiplicative
Product of objectives	$\Psi^p$	free	Definition 34	Multiplicative
Sum of objectives	$\Psi^s$	free	Definition 33	Additive
Tradeoff utility	$\Psi^t$	free	Definition 42	None
Weighted Chebyshev	$\Psi^{wc}$	dependent	Definition 36	Maximal
Weighted product	$\Psi^{wp}$	dependent	Definition 34	Multiplicative
Weighted sum	$\Psi^{ws}$	dependent	Definition 33	Additive

Table 3.1.: Overview of parameter dependency of the scalarization functions presented in Section 3.1.

ization function for which objectives are transformed by multiplying them by non-negative weights. Each weight signifies the relative importance of every corresponding objective. Weights can also be used to normalize objectives to a common numeric scale for better comparability, although it is usually recommended to decouple normalization and importance weighting [GEF16]. The popularity of the weighted sum method can be attributed to the simplicity of its application and the understanding of the meaning of the individual weights [MA04]. In changing individual weights, different regions of the Pareto front become more preferable [Mie99]. If all weights are set to the same positive number, the weighted sum is equivalent to the sum of objectives. The sum of objectives represents the case, in which all objectives are equally important to the DM.

**Definition 33** (weighted sum and sum of objectives cf. [Mie99]). *Let  $\lambda \in \mathbb{R}_{\geq 0}^m$  be a vector of non-negative weights. The weighted sum is given by*

$$\Psi^{ws}(\mathbf{u}) := \sum_{i \in [m]} \lambda_i u_i \quad (3.4)$$

and the sum of objectives is defined as

$$\Psi^s(\mathbf{u}) := \sum_{i \in [m]} u_i. \quad (3.5)$$

#### 3.1.5. Weighted Product and Product of Objectives

The weighted product (see Definition 34) is a multiplicative scalarization function that follows a similar line of thought as the weighted sum. Individual

### 3. Scalarized Preferences

objectives are exponentiated by a non-negative number to express a different priority for each goal. In contrast to the weighted sum, the weighted product is less prevalent in MOO. Describing the relative importance of each objective in terms of weights is deemed more difficult for the weighted product compared to the weighted sum [MA04]. Nonetheless, the weighted product is a popular method for describing preferences in economics, within which it is commonly referred to as Cobb-Douglas preference [CD28, VR10]. In case all weights possess the same positive value,  $\Psi^{wp}$  becomes the product of objective values.

**Definition 34** (weighted product and product of objectives [Bri22]). *Let  $\lambda \in \mathbb{R}_{\geq 0}^m$  be a vector of non-negative weights. The weighted product is given by*

$$\Psi^{wp}(\mathbf{u}) := \prod_{i \in [m]} u_i^{\lambda_i} \quad (3.6)$$

and the product of objectives is defined as

$$\Psi^P(\mathbf{u}) := \prod_{i \in [m]} u_i. \quad (3.7)$$

#### 3.1.6. Chebyshev Method

There also exist instances of maximal scalarization functions that utilize weights. Objective values can be multiplied by non-negative weights and the maximum of all products constitutes the scalarization value. A scalarization function that builds on this approach, and which is frequently used in MOO [LZ09, Mie99, MA04], is the weighted Chebyshev method (see Definition 36) that additionally uses the ideal point of the MOOP to solve. The ideal point is the vector of the smallest values that can be achieved in each objective (see Definition 35 and Figure 3.2). The weighted Chebyshev method subtracts the ideal point from the vector  $\mathbf{u}$ , whose scalarization value is computed, and multiplies the difference by a non-negative weight. The maximum of all weighted differences constitutes the scalarization value. In subtracting the ideal point from  $\mathbf{u}$ , the origin of  $\mathbb{R}^m$  becomes the joint point of reference for each objective – the smallest value that can be achieved by every transformed objective. For any Pareto optimal point  $\mathbf{u}$ , there exists a  $\lambda \in \mathbb{R}_{\geq 0}^m$  such that  $\mathbf{u}$  is the solution to  $\Psi^{wc}$  [Mie99]. Thereby, using the weighted Chebyshev method, any point of the Pareto front can be attained as preferred solution.

**Definition 35** (ideal point cf. [Mie99]). *Let  $f = (X, Y, F)$  be the objective function of an MOOP. The ideal point of  $f$  is defined as  $\mathbf{u}^*$  with  $\forall i \in [m] : u_i^* = \min_{\mathbf{u} \in \mathcal{Y}} u_i$ .*

### 3.1. The Scalarized Preference Model

**Definition 36** (Chebyshev method cf. [Mie99]). Let  $\lambda \in \mathbb{R}_{\geq 0}^m$  be a vector of non-negative weights. The weighted Chebyshev method is given by

$$\Psi^{wc}(\mathbf{u}) := \max_{i \in [m]} (\lambda_i (u_i - u_i^*)) \quad (3.8)$$

and the Chebyshev method is defined as

$$\Psi^c(\mathbf{u}) := \max_{i \in [m]} (u_i - u_i^*). \quad (3.9)$$

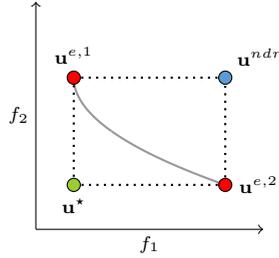


Figure 3.2.: Illustration of the ideal point  $\mathbf{u}^*$ , the nadir point  $\mathbf{u}^{ndr}$  and the extreme points for objective one ( $\mathbf{u}^{e,1}$ ) and objective two ( $\mathbf{u}^{e,2}$ ).

Sum, product and maximum constitute three fundamentally different approaches to representing preferences. This circumstance is best illustrated by assessing the indifference curves of the sum of objectives, product of objectives and Chebyshev method (see Figure 3.3). The indifference curve of a scalarization function for a given aspiration level  $a \in \mathbb{R}$  comprises all elements of the objective space that have the same scalarization value  $a$  (see Definition 37). All points that lie on the same indifference curve are thereby equally desirable. Indifference curves are a popular tool in economics for comparing different preference models [VR10]. In MOO, this concept can be used to assess, whether a move in the objective space in a given direction would be beneficial to the DM.

**Definition 37** (indifference curve cf. [Edg81]). Let  $\Psi$  be a scalarization function and  $a \in \mathbb{R}$  be an aspiration level. The indifference curve of  $\Psi$  at  $a$  is given as the implicit function

$$I(\mathbf{u}, a) := \Psi(\mathbf{u}) - a = 0. \quad (3.10)$$

Scalarization functions that involve weights enable the DM to prioritize individual objectives. Choosing the right weights, however, is usually cumbersome for a DM, which is why there exists abundant literature on MCDA methodology for providing assistance during this process [GEF16]. The subsequent

### 3. Scalarized Preferences

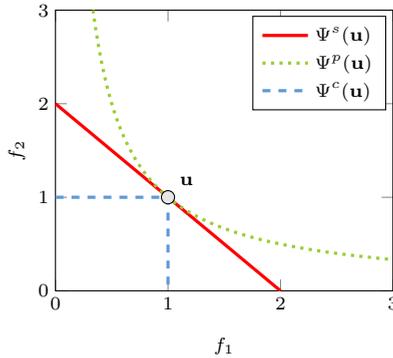


Figure 3.3.: Comparison of indifference curves for the sum of objectives, product of objectives and the Chebyshev method with  $\mathbf{u}^* = (0, 0)$ . An aspiration level of  $a = 1$  is depicted in the plot.

scalarization functions that are discussed in this section have been specifically designed to be parameter-free, instead of being mere instances of parameter-dependent scalarization functions. These functions aim at striking a balance between the different objectives by considering additional information that can be derived from the composition of the feasible set of the objective space. The Chebyshev method – although being an instance of a parameter-dependent scalarization function – follows the same strategy by utilizing the ideal point, which consists of the minima of the respective feasible objective values.

#### 3.1.7. Nash Bargaining Solution

The Nash bargaining solution (see Definition 39) is a concept that originated in axiomatic bargaining theory in welfare economics and it is an instance of a multiplicative scalarization function. The point  $\mathbf{u}$ , whose scalarization function is computed, is subtracted from the vector of the worst objective values among all Pareto optimal points – the nadir point (see Definition 38 and Figure 3.2). The differences are multiplied and the resulting product is negated. The Nash bargaining solution deems points located close to the boundary of the Pareto front as undesirable, because individual objective values that are close to their nadir value enter Equation (3.11) as small factors rendering  $\Psi^n$  large in turn (see Figure 3.4).

**Definition 38** (nadir point [CCLVV07, Deb01]). *Let  $f = (X, Y, F)$  be the objective function of an MOOP. The nadir point of  $f$  is defined as  $\mathbf{u}^{ndr}$  with  $\forall i \in [m] : u_i^{ndr} = \max_{\mathbf{u} \in \mathcal{Y}_p} u_i$ .*

### 3.1. The Scalarized Preference Model

**Definition 39** (Nash bargaining solution cf. [Nas50]). *The Nash bargaining solution is given by*

$$\Psi^n(\mathbf{u}) := - \prod_{i \in [m]} (u_i^{ndr} - u_i). \quad (3.11)$$

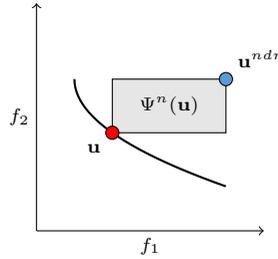


Figure 3.4.: Geometric interpretation of the scalarization value computed by the Nash bargaining solution. The area of the shaded rectangle is equivalent to the Nash bargaining value of  $\mathbf{u}$ . A larger area indicates a higher desirability.

The Nash bargaining solution was originally proposed for maximization problems and has been adapted for minimization as proposed in [BHSS17]. Subtracting objective values from the corresponding nadir values constitutes a strictly decreasing transformation, which is why the resulting product is negated.

#### 3.1.8. Angle Utility

Angle utility (see Definition 41 and Figure 3.6) is a scalarization function that computes the scalarization value of a point  $\mathbf{u}$  by using the extreme points of the Pareto front as reference. Extreme points (see Definition 40 and Figure 3.2) mark the corners of the Pareto front. They are worst in at least one objective with respect to the entire Pareto front and can thereby be deemed least interesting. Note that there exist different approaches to defining extreme points for three and more objectives [DJ14, SIR11]. Angle utility has been originally introduced as utilizing extreme points obtained by minimizing an achievement scalarization function (ASF) [BSS17, DJ14]. The ASF is a maximal scalarization function that multiplies each objective by a positive weight. If the extreme point of objective  $k$  is sought, the weight of objective  $k$  is chosen to be much smaller in comparison to the weights of the other objectives. Thereby, the optimizer of the ASF attains a large value in objective  $k$ , while the objective values of other objectives are small. Thereby, the optimizer is *extreme* in objective  $k$ .

### 3. Scalarized Preferences

In this work, a lexicographical extreme point definition is used instead (Definition 40 and Figure 3.5), since there exists a unique lexicographical extreme point in each objective [Ehr05] and minimizing an ASF might yield multiple solutions. A continuous Pareto front in three or higher dimensions might even exhibit an infinite number of ASF extreme points as shown in Example 2. Lexicographical extreme points are obtained by minimizing the objectives of an MOOP lexicographically. The extreme point of objective  $k$  is obtained by minimizing objective  $k + 1$  first. Subsequently, objective  $k + 2$  is minimized while adding the constraint that objective  $k + 1$  retains its globally optimal value. This procedure is repeated until finally objective  $k$  is minimized. The resulting point has the maximum value in objective  $k$  among all Pareto optimal points. Note that the objectives can be minimized in any order to obtain valid extreme points, as long as objective  $k$  is minimized last. However, changing the order may result in obtaining different extreme points.

**Example 2.** Let  $\mathcal{Y}_p := \{\mathbf{u} \in [0, 1]^3 \mid u_1 = 1 - u_2, u_3 = 0\}$  and  $\varepsilon > 0$  with  $\boldsymbol{\lambda} = (1/\varepsilon, 1/\varepsilon, \varepsilon)$ . It follows that  $\boldsymbol{\lambda}\mathbf{u}$  is constant for any  $\mathbf{u} \in \mathcal{Y}_p$ , which makes all  $\mathbf{u} \in \mathcal{Y}_p$  the extreme point of objective 3.

Using angle utility, desirability is measured by considering the geometric properties of the front, namely the angles between  $\mathbf{u}$  and the extreme points (see Figure 3.6). The largest of all these angles yields the scalarization value of  $\mathbf{u}$ . This way, angle utility favors points that are located either towards the center of the front or at convex bulges, since such points tend to decrease the angles computed in Equation (3.14) (see Figure 3.6). Although angle utility utilizes the maximum function (see Equation (3.13)) to compute scalarization values, it is not a maximal scalarization function, since each angle is computed using the entire vector  $\mathbf{u}$  instead of a single component of  $\mathbf{u}$  (see Equation (3.14)). This implies an interdependency in the valuation of individual objectives that is not considered in the maximal scalarization function.

**Definition 40** (extreme point cf. [Ehr05]). Let  $f = (X, Y, F)$  be the objective function of an MOOP. The  $k$ -th extreme point  $\mathbf{u}^{e,k}$  of  $f$  is given by

$$\mathbf{u}^{e,k} := \operatorname{lexmin}_{\mathbf{u} \in \mathcal{Y}}(u_{k+1}, \dots, u_m, u_1, \dots, u_k). \quad (3.12)$$

**Definition 41** (angle utility [BSS17]). The angle utility is given by

$$\Psi^a(\mathbf{u}) = \max_{k \in [m]} (\gamma(\mathbf{u}, \mathbf{u}^{e,k})), \quad (3.13)$$

where

$$\gamma(\mathbf{u}, \mathbf{u}^{e,k}) = \arctan \left( \frac{\sqrt{\sum_{i=1, i \neq k}^m (u_i - u_i^{e,k})^2}}{|u_k - u_k^{e,k}|} \right). \quad (3.14)$$

### 3.1. The Scalarized Preference Model

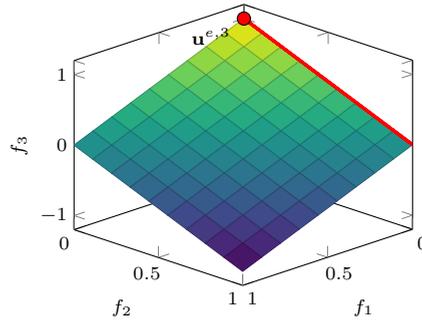


Figure 3.5.: Illustration of lexicographic extreme points for three objectives on the Pareto front  $\mathcal{Y}_p := \{\mathbf{u} \in [-1, 1]^3 \mid (u_1, u_2) \in [0, 1]^2 : u_3 = 1 - u_1 - u_2\}$ . The extreme point of objective three is obtained by minimizing objective one first resulting in the set  $\{(0, u_2, u_3) \in [0, 1]^3 \mid u_2 - u_3 = 1\}$  indicated by the thick red line. Further minimizing objective two yields  $\mathbf{u}^{e,3}$ .

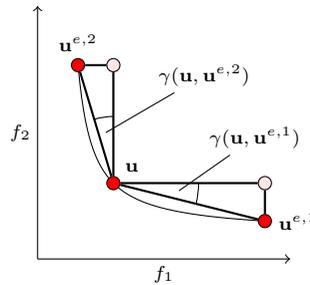


Figure 3.6.: Illustration of the computation of angle utility.

#### 3.1.9. Tradeoff Utility

Tradeoff utility is as scalarization function that is based on the concept of tradeoff optimality that has been introduced in Section 2.3 (see Definitions 14 and 42). Instead of considering only the tradeoff to one or multiple reference points (see Definitions 36, 39 and 41), tradeoffs are measured with respect to all elements of the feasible set of the objective space  $\mathcal{Y}$ . The tradeoff between two points is computed as the fraction of the worst deterioration across all objectives divided by the best improvement across all objectives. The maximum tradeoff of a point to all other elements of  $\mathcal{Y}$  constitutes its tradeoff utility. In this sense, the desirability of a point is determined by the best alternative available. Although tradeoff utility possesses properties of a maximal scalarization function, it can-

### 3. Scalarized Preferences

not be characterized as such, since individual objectives are not translated by monotonous transformations.

**Definition 42** (tradeoff utility [SBS13]). *The tradeoff utility is given as*

$$\Psi^t(\mathbf{u}) := \max_{\mathbf{v} \in \mathcal{Y}} \frac{\max_{i \in [m]} (u_i - v_i)}{\max_{i \in [m]} (v_i - u_i)}. \quad (3.15)$$

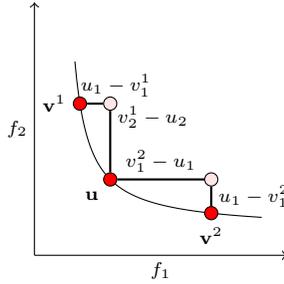


Figure 3.7.: Illustration of tradeoff utility. The fraction in Equation (3.15) is computed by dividing the length of the corresponding lines indicated in the figure. The numerator of said fraction is the greatest loss among all objective values that occurs when choosing  $\mathbf{u}$  instead of  $\mathbf{v}^1$ , which corresponds to  $u_1 - v_1^1$ . The denominator represents the largest gain when selecting  $\mathbf{u}$  instead of  $\mathbf{v}^1$ , which corresponds to  $v_1^1 - u_2$ . The same observation applies to  $\mathbf{v}^2$ .

## 3.2. Characterizing Scalarization Functions

The scope of this section is the mathematical characterization of scalarization functions using a system of axioms. These axioms represent desirable properties that a scalarization function can exhibit. A DM may deem some axioms more important than others. In providing an axiomatic characterization, a DM can be supported in identifying a scalarization function that serves as suitable representation of her preferences. Depending on the set of axioms a scalarization function fulfills, different optimization methodologies may be applied to obtain preferred solutions. The lack of fulfilling certain axioms, on the other hand, may necessitate the modification of an existing optimization algorithm before it can be combined with a given scalarization function (cf. [SBS13, BHSS17]).

The majority of the axioms that are presented in this section are founded in welfare economics and social choice theory. Welfare economics is concerned with

### 3.2. Characterizing Scalarization Functions

the optimal allocation of goods in a society with multiple individuals, while social choice theory, more generally, deals with collective decision making in maximizing the well-being of a society. Depending on the preferences of the members of a society, a hypothetical social planner is tasked with distributing goods in such a manner that social welfare – the overall well-being of the members – is maximized. Individual utility derived from the allocation of goods is aggregated using social welfare functionals (SWFs). These functionals are characterized in a systematic manner by axioms – desirable properties that a distribution mechanism may or should exhibit [ASS02, Roe98].

Welfare economics and social choice theory have been mainly acknowledged in MCDA and MOO in group decision making [CCLVV07, GEF16], however, both disciplines have been scarcely related to scalarization in an optimization context [Sta79]. Such a connection makes sense on a conceptual level, since a social planner trying to find an allocation of goods that satisfies members of a society can be related to a DM aiming to obtain a solution that strikes a balance between the different objectives. The different allocations of goods, among which a social planner can choose, correspond to the elements of the feasible set in MOO. Individuals in welfare economics may be interpreted as the objectives being optimized. Utility functions, employed by individuals to evaluate allocations, are objective functions in MOO.

Although SWFs and scalarization functions have so far not been explicitly interrelated, there exist SWFs that have been employed as scalarization functions to obtain preferred solutions in MOO – sometimes under different names. The sum of objectives and the weighted sum are known as utilitarian and weighted utilitarian SWF in welfare economics [dG02]. The Nash bargaining solution, although not prevalent in MOO, has been used to obtain preferred solutions to MOOPs [MMD91, BSS15, BSS17]. The maximal scalarization function corresponds to the egalitarian SWF (or Rawlsian SWF) [Raw71].

It should be noted that other parts of economic theory that are related to welfare theory, such as utility theory, have been largely acknowledged and applied in relation to scalarization functions [Mie99, Bel86, HR88]. Utility theory is concerned with the representation of human preferences by real-valued functions [VNM53], and thereby builds the basis of MAUT and MAVT methodology [KR93]. The crucial difference between utility theory and welfare economics with respect to decision making is the perspective from which both disciplines approach the subject. Utility theory takes a positive stance in assuming that the DM inherently possesses a preference and aims at finding a numerical representation of it [VNM53]. Welfare and social choice theory follow a normative perspective in defining characteristics a decision making process should adhere to [ASS02]. While the positive perspective has been explored, largely in MCDA

### 3. Scalarized Preferences

[GEF16], partly in MOO [Mie99], the normative perspective has not been sufficiently considered. In developing an axiomatic characterization of scalarization functions, this work aims at establishing a framework for a normative description of scalarization functions.

Axioms to characterize SWFs are guided by ideas of moral philosophy and distributive justice [dG02, Roe98]. Consequently, not all considerations in welfare economics and social choice theory translate well to MOO. After all, objectives are not humans. For this reason, each axiom must be carefully assessed for its applicability in MOO. Many axioms have been proposed in welfare economics, since its inception by Kenneth Arrow [Arr51]. A review of all these characterizations is beyond the scope of this thesis. Instead, an emphasis is put on a selected few of these axioms that bear a significant meaning in the optimization context.

In social choice theory, each individual  $i$  possesses an evaluation profile that is a numeric representation of  $i$ 's preference on the set of alternatives  $X$ , among which the social planner can choose from. Each individual assigns a real value to every alternative of  $X$  signifying its desirability [Sen70]. This corresponds to an objective function measuring the fulfillment of a goal by ranking outcomes generated by choosing different values for the decision variables. The evaluation profiles of all members of a given society can be aggregated into a tuple. If there exist  $N$  individuals and  $X$  is finite the evaluation profile of the society is an  $|X| \times N$  matrix. If  $X$  is continuous, then the aggregated profile is a subset of  $\mathbb{R}^N$ . The aggregated evaluation profile of the society is the domain of the SWF [Sen70] and corresponds to the feasible set of the objective space  $\mathcal{Y}$  in MOO, which is the domain of the scalarization function. Whether a given SWF satisfies or violates a specific axiom can then either be defined for all feasible evaluation profiles or a subset of evaluation profiles [dG02]. The same line of thought is adopted in this work. The fulfillment or violation of the axioms proposed in this section is defined with respect to feasible sets of the objective space or to restrictions of these sets (see Definition 43).

**Definition 43** (feasible objective spaces and feasible Pareto fronts). *Let  $\mathcal{F}$  denote the set of all objective functions that define a real-valued MOOP. The set of feasible objective spaces is defined as*

$$\Upsilon := \{ \mathcal{Y} \subseteq \mathbb{R}^m \mid \exists f \in \mathcal{F} : \mathcal{Y} \text{ is the image of } f \}. \quad (3.16)$$

*Furthermore, the set of feasible Pareto fronts is defined as*

$$\Upsilon_p := \{ \mathcal{Y}_p \subseteq \mathbb{R}^m \mid \exists f \in \mathcal{F} : \mathcal{Y}_p \text{ is the Pareto front of } f \}. \quad (3.17)$$

### 3.2.1. Pareto Compliance

The first axiom that is presented within the proposed framework is based on the Pareto dominance principle (see Definition 7). Welfare theory distinguishes between the weak and strong Pareto axiom. The weak Pareto axiom states that if alternative  $x$  is preferred by all individuals of the society to alternative  $y$ , the SWF must rank  $x$  higher to  $y$ . The strong Pareto axiom states that if at least one individual prefers  $x$  to  $y$  and all other individuals only weakly prefer  $x$  to  $y$  (implying they either prefer  $x$  to  $y$  or they are indifferent between both alternatives), then the SWF must rank  $x$  higher than  $y$ . Strong Pareto guarantees that no inefficient alternative is chosen such that at least one individual could be made better off without making anyone else worse off [Par96, dG02].

The Pareto dominance principle forms the very basis of defining optimality in MOO (cf. Definition 8). This principle has also been applied in the context of scalarization functions in MOO [Mie99, MA04, SBS13]. Scalarization functions are usually assumed to be strictly increasing in each objective, implying that the scalarization value becomes smaller if any objective value decreases [Mie99]. In most applications, however, it is deemed sufficient if the minimization of the scalarization function yields a Pareto optimal point [MA04]. The Pareto compliance axiom proposed in this work states that any dominated alternative must possess a larger scalarization value than any alternative by which it is dominated. Any scalarization function that satisfies Pareto compliance is guaranteed to obtain a Pareto optimal point when minimized.

**Definition 44** (Pareto compliance cf. [Par96, Arr51]). *A scalarization function  $\Psi$  fulfills Pareto compliance if for all  $\mathcal{Y} \in \Upsilon$  and  $\mathbf{u}, \mathbf{v} \in \mathcal{Y}$  satisfying  $\mathbf{u} <_p \mathbf{v} \Rightarrow \Psi(\mathbf{u}) < \Psi(\mathbf{v})$ .*

Pareto compliance has further been acknowledged in MOO with respect to quality indicators. Quality indicators are functions that map a set of vectors to a real number (see Section 5.3.1.3). Pareto compliance in terms of quality indicators means that, given two sets  $A$  and  $B$ , if every element of  $B$  is dominated by at least one element of  $A$ , then the indicator value of  $B$  must be larger than the indicator value of  $A$  [CCLVV07, Definition 32].

The Pareto compliance axiom is a requirement for a scalarization function to qualify as preference predicate, which mandates the preservation of the Pareto order (see Definition 12). In case a scalarization function does not satisfy Pareto compliance, a check for Pareto dominance must be performed in addition before comparing scalarization values. This leads to an increase in runtime when executing optimization algorithms [SBS13].

**Proposition 4.** *Let  $S^P$  denote the set of scalarization functions that satisfy Pareto and let  $\mathcal{Y} \in \Upsilon$ . Then,*

### 3. Scalarized Preferences

1.  $\Psi^+, \Psi^{ws}, \Psi^s \in S^P$ ,
2.  $\Psi^* \in S^P$  if for all  $i \in [m]$  it holds that  $g_i : \mathbb{R} \rightarrow \mathbb{R}_+$ , i.e. all transformation functions map to positive numbers.
3.  $\Psi^{wp}, \Psi^p \in S^P$  if  $\mathcal{Y} \subseteq \mathbb{R}_+^m$ , i.e. the feasible objective space is fully contained in the positive orthant.
4.  $\Psi^n \in S^P$  for all  $\mathbf{u} \in \mathcal{Y}$  satisfying  $\mathbf{u} < \mathbf{u}^{ndr}$ , i.e. points that are strictly smaller in each component compared to the nadir point.
5.  $\Psi^a \in S^P$  for all  $\mathbf{u} \in \mathcal{Y}$  and  $k \in [m]$  for which there does not exist an  $\mathbf{u}^{e,k}$  such that  $\mathbf{u}^{e,k} <_p \mathbf{u}$ , i.e. points that are not Pareto dominated by any extreme point.
6.  $\Psi^{\max}, \Psi^{wc}, \Psi^c, \Psi^t \notin S^P$ .

*Proof.* Let  $\mathbf{u}, \mathbf{v} \in \mathcal{Y}$  be arbitrary and  $\mathbf{u} <_p \mathbf{v}$ .

1. If  $\mathbf{u} <_p \mathbf{v}$  it follows that for all  $i \in [m]$  that  $u_i \leq v_i$  and for at least one  $j \in [m]$  that  $u_j < v_j$ . Since the  $g_i$  are strictly increasing transformation functions, it follows that  $g_i(u_i) \leq g_i(v_i)$  and  $g_j(u_j) < g_j(v_j)$ , which implies that  $\Psi^+(\mathbf{u}) < \Psi^+(\mathbf{v})$ . It follows that for all  $k \in \{+, ws, s\}$  that  $\Psi^k \in S^P$ .
2. The scalarization function  $\Psi^*$  is strictly increasing in each objective if for all  $i \in [m]$  it holds that  $g_i : \mathbb{R} \rightarrow \mathbb{R}_+$ . It follows that  $\Psi^*(\mathbf{u}) < \Psi^*(\mathbf{v})$ .
3. Since  $\Psi^* \in S^P$  if the  $g_i$  map to positive numbers, it follows that  $\Psi^{wp}, \Psi^p \in S^P$  if  $\mathcal{Y} \subseteq \mathbb{R}_+^m$ .
4. The result  $\Psi^n \in S^P$  for all  $\mathbf{u} \in \mathcal{Y}$  satisfying  $\mathbf{u} < \mathbf{u}^{ndr}$  is a consequence of  $\Psi^* \in S^P$  if for all  $i \in [m] \Rightarrow g_i : \mathbb{R} \rightarrow \mathbb{R}_+$ .
5. For a proof of  $\Psi^a \in S^P$  for all  $\mathbf{u} \in \mathcal{Y}$  and  $k \in [m]$  for which there does not exist an  $\mathbf{u}^{e,k}$  such that  $\mathbf{u}^{e,k} <_p \mathbf{u}$  see [BSS17, Proposition 1].
6. Let  $\mathbf{u}^* = (0, 0)$ ,  $\mathbf{u} = (1, 0)$  and  $\mathbf{v} = (1, 1)$ . It follows that  $\Psi^c(\mathbf{u}) = \Psi^c(\mathbf{v})$ . Consequently,  $\Psi^{\max}, \Psi^{wc}, \Psi^c \notin S^P$ .

For a proof of  $\Psi^t \notin S^P$  see [SBS13, Theorem 3].

□

Proposition 4 shows that most of the scalarization functions presented in Section 3.1 satisfy Pareto compliance only if the feasible set of the objective space is restricted. These restrictions differ for each scalarization function and can be remedied by applying different actions. The weighted product and the product of objectives are only strictly increasing within the positive orthant. If objective

### 3.2. Characterizing Scalarization Functions

values become smaller than zero, an objective vector  $\mathbf{u}$  may Pareto dominate other points that possess smaller scalarization values than  $\mathbf{u}$  – e.g.  $\mathbf{u} = (-5, -5)$ ,  $\mathbf{v} = (1, 1)$ . This observation also illustrates, why the transformation functions of any multiplicative scalarization function must strictly map to the positive orthant for the scalarization function to fulfill the Pareto compliance axiom.

In many real-world optimization problems objective values take on negative values [BDMS16, BSE<sup>+</sup>16]. Negative objective values can be scaled to positive values by subtracting a number greater than the smallest attainable value of this objective from the corresponding objective function. Such a transformation, however, requires knowledge of the range of values that the objectives can take on. Such knowledge is often not available prior to any optimization effort [MA04]. Scaling may also effect the position of the scalarization optimum, an aspect that is later discussed under invariance to scale axioms.

The scalarization value computed by the Nash bargaining solution is equivalent to the volume of the box encompassed by  $\mathbf{u}$  and the nadir point (Figure 3.4). A larger volume is associated with a higher desirability. This notion, however, makes only sense as long as  $\mathbf{u}$  dominates the nadir point in each objective. For example, if the nadir point dominates  $\mathbf{u}$ , moving  $\mathbf{u}$  towards positive infinity increases the volume but not the desirability of  $\mathbf{u}$ .

The angles utilized for the computation of angle utility are an indicator for the tradeoff between  $\mathbf{u}$  and the extreme points of the Pareto front. In this context, the  $k$ -th angle may be perceived as the tradeoff with respect to the  $k$ -th objective. Angles are a meaningful notion for measuring the tradeoff as long as  $\mathbf{u}$  possesses a smaller objective value than the extreme point of the  $k$ -th objective. If  $\mathbf{u}^{e,k}$  dominates  $\mathbf{u}$  then no tradeoff occurs as  $\mathbf{u}^{e,k}$  is strictly better than  $\mathbf{u}$ . The  $k$ -th angle loses its meaning as indicator for measuring tradeoffs in this situation.

The weighted Chebyshev method can be modified to fulfill the Pareto compliance axiom: Given a  $\mathbf{u} \in \mathcal{Y}$ , the weighted Chebyshev method can be augmented by summing the components of  $\mathbf{u} - \mathbf{u}^*$ , multiplying the sum by a small positive scalar and adding the result to  $\Psi^{wc}$  [MA04]. The Nash bargaining solution and angle utility can only be used to rank points that dominate the Nadir point. In using intermediate estimates of the Nadir point and the extreme points, however, it is still possible to design optimization algorithms that obtain the corresponding scalarization optima when starting the search from an arbitrary position in the search space [BSS15, BSS17]. There exist no results that show under which conditions Pareto compliance holds for tradeoff utility. Existing algorithms that utilize tradeoff utility therefore additionally perform Pareto domination checks when comparing points in the objective space [SBS13].

### 3. Scalarized Preferences

#### 3.2.2. Binary Independence

SWFs are also characterized by the amount of information they require about the set of alternatives  $X$  to establish a social preference between two alternatives  $x$  and  $y$ . The axiom of binary independence<sup>1</sup> states that the social preference between  $x$  and  $y$  depends only on the individuals' preferences between  $x$  and  $y$  and no other alternatives in  $X$ . The social ranking generated by SWFs that violate binary independence depends on the composition of  $X$ . Adding or removing alternatives to  $X$  can thereby change the social preference of  $x$  to  $y$  [Arr51]. Binary independence, however, has also been criticized in the economics literature, since human preferences are known to depend on the information available. The revelation of additional choices can thereby lead to a reevaluation of existing options [KT79].

The notion of binary independence is directly transferable to MOO by assessing if the computation of scalarization values requires any knowledge about the composition of the feasible set of the objective space  $\mathcal{Y}$ . Such knowledge often consists of reference points such as the ideal or nadir point (see Definitions 36 and 39). Binary independence in the context of scalarization implies that the preference between two points  $\mathbf{u}, \mathbf{v} \in \mathcal{Y}$  is independent of the composition of  $\mathcal{Y}$  (see Definition 45). If a scalarization function requires knowledge about the composition of  $\mathcal{Y}$ , that knowledge must be known to correctly compute scalarization value. In MOO, however, there is usually no information available about  $\mathcal{Y}$  prior to any optimization effort. This implies, in turn, that scalarization optima can only be correctly identified, if the necessary knowledge about  $\mathcal{Y}$  is obtained first.

The axiom of binary independence is formulated using the mathematical concept of restrictions of functions. The restriction of a function consists of truncating its domain, while maintaining all existing relations defined by its graph of the elements of the domain that have not been removed. A scalarization function fulfills binary independence if it establishes the same ranking between two points on all of its restrictions of which these two points form part.

**Definition 45** (binary independence cf. [Arr51]). *A scalarization function  $\Psi$  fulfills binary independence if for all  $\mathcal{Y}_1, \mathcal{Y}_2 \in \Upsilon$  for which  $\mathbf{u}, \mathbf{v} \in \mathcal{Y}_1$  and  $\mathbf{u}, \mathbf{v} \in \mathcal{Y}_2$  and  $R \in \{<, =, >\}$  it follows that  $\Psi|_{\mathcal{Y}_1}(\mathbf{u}) R \Psi|_{\mathcal{Y}_1}(\mathbf{v}) \Rightarrow \Psi|_{\mathcal{Y}_2}(\mathbf{u}) R \Psi|_{\mathcal{Y}_2}(\mathbf{v})$ .*

The requirement of additional information imposes different restriction on the applicability of the scalarized preference model (see Proposition 5). While the ideal point can be obtained by minimizing each objective function individually, computing the nadir point is known to be a difficult problem for three and

---

<sup>1</sup>Binary independence is also called independence of irrelevant alternatives.

### 3.2. Characterizing Scalarization Functions

more objectives [CCLVV07]. Computing the extreme points of a Pareto front is equally difficult as determining its nadir point, since the  $i$ -th objective of the  $i$ -th extreme point is the  $i$ -th component of the nadir point. A single misidentified extreme point may lead to a large perturbation in the computed scalarization values making the correct identification of the scalarization optimum impossible [BSS17]. Computing the exact tradeoff utility of a single point consists of solving an optimization on its own (see Definition 42). Although there exist simplified formulations to solving Equation (3.15) for two objective problems, depending on the curvature of the Pareto front,  $\mathcal{Y}_p$  must be known for computing tradeoff utility exactly for arbitrary MOOPs [SBS13]. Despite these issues, it has been shown, that the necessary knowledge about  $\mathcal{Y}$  for computing scalarization values of the Chebyshev method, the Nash bargaining solution, angle utility and tradeoff utility can be estimated and obtained during runtime to approximate scalarization optima [SBS13, BSS15, BSS17, BHSS17].

**Proposition 5.** *Let  $S^{BI}$  denote the set of scalarization functions that satisfy binary independence. Then,*

1.  $\Psi^{ws}, \Psi^s, \Psi^{wp}, \Psi^p \in S^{BI}$ ,
2.  $\Psi^{wc}, \Psi^c, \Psi^n, \Psi^a, \Psi^t \notin S^{BI}$ .

*Proof.* Let  $i \in \{ws, s, wp, p\}$ . Evidently, for all  $\mathcal{Y}_1, \mathcal{Y}_2 \in \Upsilon$  and  $\mathbf{u} \in \mathcal{Y}_1, \mathbf{u} \in \mathcal{Y}_2$  it holds that  $\Psi^i|_{\mathcal{Y}_1}(\mathbf{u}) = \Psi^i|_{\mathcal{Y}_2}(\mathbf{u})$ . Scalarization functions  $\Psi^{wc}$  and  $\Psi^c$  require the ideal point,  $\Psi^n$  the nadir point,  $\Psi^a$  the extreme points and  $\Psi^t$  necessitates the knowledge of  $\mathcal{Y}_p$  to compute scalarization values.  $\square$

In general, utilizing more knowledge about the composition of  $\mathcal{Y}$  enables the DM to make a more informed decision about the solutions she prefers. Including such knowledge in the computation of scalarization values is thereby expected to be beneficial. However, since this information has to be obtained first, computational resources have to be diverted that could otherwise be spent on the search for scalarization optima.

#### 3.2.3. Non-Extremeness

The next axiom that is discussed is founded in the nondictatorship property. Social choice theory characterizes a SWF as dictatorial if only the preferences of a single individual are decisive for the social ranking of all available alternatives. Such an SWF maximizes the well-being of a single individual, while neglecting the opinions of other members of the society. This is undesirable from a normative perspective, since social welfare is reduced to the well-being of only one individual [Arr51, Roe98].

### 3. Scalarized Preferences

A scalarization function that minimizes a single objective while neglecting all other objectives may be perceived as *dictatorial* in the context of MOO. Minimizing only one or several objectives is problematic if it comes at the expense of attaining values in other objectives that coincide with their corresponding nadir values. Such solutions provide no balance between the different objectives and are thereby not interesting candidates for implementation [GEF16]. The axiom of non-extremeness states that a scalarization function must avoid identifying any extreme point as scalarization optimum if the Pareto front does not solely consist of extreme points.

**Definition 46** (non-extremeness). *A scalarization function  $\Psi$  fulfills non-extremeness if for all  $\mathcal{Y}_p \in \Upsilon_p$  and  $k \in [m]$  for which  $\mathcal{Y}_p \setminus \{\mathbf{u}^{e,1}, \dots, \mathbf{u}^{e,m}\} \neq \emptyset$  it follows that*

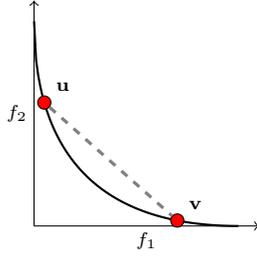
$$\arg \min_{\mathbf{u} \in \mathcal{Y}_p} \Psi(\mathbf{u}) \neq \mathbf{u}^{e,k}. \quad (3.18)$$

Whether a scalarization function identifies an extreme point as optimum, mainly depends on the curvature of the Pareto front (Figure 3.8). It is a well-known fact in MOO that the weighted sum method identifies extreme points as optima on strictly concavely shaped Pareto fronts irrespective of the chosen weights [DD97, MPSM00]. Proposition 6 shows that this result can be generalized for additive scalarization functions. For any additive scalarization function it is possible to construct a concave Pareto front on which the scalarization function identifies extreme points as optima. Tradeoff utility also identifies extreme solutions on concave Pareto fronts as optima, since the improvement in at least one objective is always greater than the worst deterioration among all other objectives when moving from an interior point towards the boundary [SBS13]. The shape of the Pareto front, however, is usually not known prior to any optimization effort unless all objective functions and constraints are convex, in which case the Pareto front is also convex [Mie99]. Therefore, obtaining an extreme solution cannot be precluded when applying a scalarization function that violates non-extremeness.

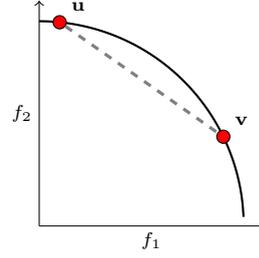
**Definition 47** (symmetric extreme points). *Let  $\mathcal{Y}_p \in \Upsilon_p$  and  $a, b \in \mathbb{R}$  with  $a < b$ . The extreme points of  $\mathcal{Y}_p$  are symmetric if for all  $k \in [m]$  and  $i \in [m] \setminus \{k\}$  it holds that  $e_i^{e,k} = a$  and  $u_k^{e,k} = b$ .*

The analysis of angle utility in the context of non-extremeness makes use of the Bachmann-Landau notation for describing the limiting behavior of functions. The limit of angle utility when moving from the interior to the extreme points of the Pareto front is assessed to analyze conditions that make angle utility fulfill non-extremeness. Symmetric extreme points (Definition 47) are a requirement for angle utility satisfying non-extremeness.

### 3.2. Characterizing Scalarization Functions



(a) Convex Pareto front.



(b) Concave Pareto front.

Figure 3.8.: Example of a convex (Figure 3.8a) and concave (Figure 3.8b) Pareto front. The line connecting any two points  $\mathbf{u}, \mathbf{v} \in \mathcal{Y}_p$  lies above the Pareto front if it is convex and below if it is concave.

**Proposition 6.** Let  $S^{NE}$  denote the set of scalarization functions that satisfy non-extremeness. Then,

1.  $\Psi^c, \Psi^n \in S^{NE}$ ,
2.  $\Psi^{wc} \in S^{NE}$  if  $\lambda \in \mathbb{R}_+^m$ ,
3.  $\Psi^a \in S^{NE}$  if the following two conditions are satisfied:
  - a) The extreme points of  $\mathcal{Y}_p$  are symmetric.
  - b) For every  $k \in [m]$  there exists a parametric equation of  $\mathcal{Y}_p$  denoted by  $h = (h_1, \dots, h_m) : \mathbb{R}^{m-1} \rightarrow \mathbb{R}^m$  for which it holds that

$$\lim_{\mathbf{t} \rightarrow \infty} h(\mathbf{t}) = \mathbf{u}^{e,k}. \quad (3.19)$$

Furthermore, there exists an  $i \in [m]$  such that for all  $j \in [m] \setminus \{i\}$  it holds that  $h_i(t) \in \omega(h_j(t))$ , i.e. there exists a component  $i$  in which  $h$  grows asymptotically faster than any other component  $j$ .

4.  $\Psi^{ws}, \Psi^s, \Psi^*, \Psi^{wp}, \Psi^p, \Psi^t \notin S^{NE}$  and
5.  $\Psi^+ \notin S^{NE}$ .

*Proof.*

1. For any Pareto optimal point  $\mathbf{u}$  there exists a weight vector  $\lambda$ , such that  $\mathbf{u}$  is the solution to  $\Psi^{wc}$ . The solution  $\mathbf{u}$  is only a boundary point if at least one  $\lambda_i = 0$  [Mie99, Theorem 3.4.2]. It follows that  $\Psi^{wc}, \Psi^c \in S^{NE}$ .

### 3. Scalarized Preferences

2. For all  $k \in [m]$  it follows that  $\Psi^n(\mathbf{u}^{e,k}) = 0$ . For any point  $\mathbf{u} \in \mathcal{Y}_p$  that does not possess a nadir value in any of its objectives it holds that  $\Psi(\mathbf{u}) < 0$ . It follows that  $\Psi^n \in S^{NE}$ .
3. For  $\Psi^a$ , the case  $m = 2$  is considered first. Since  $\gamma(\mathbf{u}^{e,k}, \mathbf{u}^{e,k})$  results in an undefined mathematical expression, the limit of  $\gamma(\mathbf{u}^{e,k}, \mathbf{u}^{e,k})$  is considered and it is shown that said limit exists. Next, it is proven that there exists an interior point that has a smaller scalarization value than the limit if the extreme points are symmetric concluding that angle utility fulfills non-extremeness.

Without loss of generality, let  $\mathbf{u}^{e,1} = (1, 0)$  and  $\mathbf{u}^{e,2} = (0, 1)^2$  and  $h(t) := (1 - h_1(t), h_2(t))$  with  $i \in \{1, 2\} : \lim_{t \rightarrow \infty} h_i(t) = 0$ . It follows that

$$\Psi^a(h(t)) = \max(\gamma(h(t), \mathbf{u}^{e,1}), \gamma(h(t), \mathbf{u}^{e,2})), \quad (3.20a)$$

$$\gamma(h(t), \mathbf{u}^{e,1}) = \arctan\left(\frac{h_2(t)}{h_1(t)}\right), \quad (3.20b)$$

$$\gamma(h(t), \mathbf{u}^{e,2}) = \arctan\left(\frac{1 - h_1(t)}{1 - h_2(t)}\right). \quad (3.20c)$$

The following cases can be distinguished:

- a) If  $h_2 \in w(h_1)$  it holds that

$$\gamma(h(t), \mathbf{u}^{e,1}) \xrightarrow{t \rightarrow \infty} \pi/2, \quad (3.21a)$$

$$\gamma(h(t), \mathbf{u}^{e,2}) \xrightarrow{t \rightarrow \infty} \pi/4, \quad (3.21b)$$

$$\Psi^a(h(t)) \xrightarrow{t \rightarrow \infty} \pi/2. \quad (3.21c)$$

Furthermore, there exist  $t_0, \varepsilon \in \mathbb{R}_+$  such that

$$\gamma(h(t_0 + \varepsilon), \mathbf{u}^{e,1}) > \gamma(h(t_0), \mathbf{u}^{e,1}) > \gamma(h(t_0), \mathbf{u}^{e,2}) > \gamma(h(t_0 + \varepsilon), \mathbf{u}^{e,2}), \quad (3.22)$$

which implies that  $\Psi^a(h(t_0)) < \Psi^a(h(t_0 + \varepsilon))$ .

- b) If  $h_1 \in \omega(h_2)$  it holds that

$$\gamma(h(t), \mathbf{u}^{e,1}) \xrightarrow{t \rightarrow \infty} 0, \quad (3.23a)$$

$$\gamma(h(t), \mathbf{u}^{e,2}) \xrightarrow{t \rightarrow \infty} \pi/4, \quad (3.23b)$$

$$\Psi^a(h(t)) \xrightarrow{t \rightarrow \infty} \pi/4. \quad (3.23c)$$

---

<sup>2</sup>See Proposition 10 for a justification of this simplification.

### 3.2. Characterizing Scalarization Functions

Furthermore, there exist  $t_0, \varepsilon \in \mathbb{R}_+$  such that

$$\gamma(h(t_0 + \varepsilon), \mathbf{u}^{e,2}) > \gamma(h(t_0), \mathbf{u}^{e,2}) > \gamma(h(t_0), \mathbf{u}^{e,1}) > \gamma(h(t_0 + \varepsilon), \mathbf{u}^{e,1}), \quad (3.24)$$

which implies that  $\Psi^a(h(t_0)) < \Psi^a(h(t_0 + \varepsilon))$ .

- c) The case  $h_1 \in \Theta(h_2)$  is in violation to the condition that either  $h_1 \in w(h_2)$  or  $h_2 \in w(h_1)$  must hold, which is why it can be neglected. For the sake of completeness, the limiting behavior of angle utility in this case is given as well.

$$\lim_{t \rightarrow \infty} \gamma(h(t), \mathbf{u}^{e,1}) \in (0, \pi/2), \quad (3.25a)$$

$$\gamma(h(t), \mathbf{u}^{e,2}) \stackrel{t \rightarrow \infty}{=} \pi/4, \quad (3.25b)$$

$$\Psi^a(h(t)) \stackrel{t \rightarrow \infty}{=} \max \left( \lim_{t \rightarrow \infty} \gamma(h(t), \mathbf{u}^{e,1}), \frac{\pi}{4} \right). \quad (3.25c)$$

Let  $h(\mathbf{t}) := (h_1(\mathbf{t}), \dots, h_{k-1}(\mathbf{t}), 1 - h_k(\mathbf{t}), h_{k+1}(\mathbf{t}), \dots, h_m(\mathbf{t}))$ . Furthermore, let  $h_{-k}$  denote the function  $h$  without its  $k$ -th index, i.e.  $h_{-k}(\mathbf{t}) = (h_1(\mathbf{t}), \dots, h_{k-1}(\mathbf{t}), h_{k+1}(\mathbf{t}), \dots, h_m(\mathbf{t}))$ . For any  $k \in [m]$  and  $i \in [m] \setminus \{k\}$  it holds that:

$$\gamma(h(\mathbf{t}), \mathbf{u}^{e,k}) = \frac{\|h_{-k}(\mathbf{t})\|_2}{h_k(\mathbf{t})}, \quad (3.26a)$$

$$\gamma(h(\mathbf{t}), \mathbf{u}^{e,i}) = \frac{\sqrt{(1 - h_k(\mathbf{t}))^2 + \sum_{j \in [m] \setminus \{i,k\}} h_j(\mathbf{t})^2}}{1 - h_i(\mathbf{t})} \quad (3.26b)$$

The following two cases can be distinguished:

- a) If  $\forall i \in [m] \setminus \{k\} : h_k \in w(h_i)$  : then

$$\lim_{\mathbf{t} \rightarrow \infty} \gamma(h(\mathbf{t}), \mathbf{u}^{e,k}) = 0, \quad (3.27a)$$

$$\lim_{\mathbf{t} \rightarrow \infty} \gamma(h(\mathbf{t}), \mathbf{u}^{e,i}) = \pi/4 \quad (3.27b)$$

Furthermore, there exist an  $\varepsilon \in \mathbb{R}_+$  and a  $\mathbf{t}_0 \in \mathbb{R}^{m-1}$  such that

$$\gamma(h(\mathbf{t}_0 + \varepsilon), \mathbf{u}^{e,i}) > \gamma(h(\mathbf{t}_0), \mathbf{u}^{e,i}) > \gamma(h(\mathbf{t}_0), \mathbf{u}^{e,k}) > \gamma(h(\mathbf{t}_0 + \varepsilon), \mathbf{u}^{e,k}), \quad (3.28)$$

which implies that  $\Psi^a(h(\mathbf{t}_0)) < \Psi^a(h(\mathbf{t}_0 + \varepsilon))$ .

- b) If  $\exists i \in [m] \setminus \{k\} : h_i \in w(h_k)$  : then

$$\lim_{\mathbf{t} \rightarrow \infty} \gamma(h(\mathbf{t}), \mathbf{u}^{e,k}) = \pi/2, \quad (3.29a)$$

$$\lim_{\mathbf{t} \rightarrow \infty} \gamma(h(\mathbf{t}), \mathbf{u}^{e,i}) = \pi/4 \quad (3.29b)$$

### 3. Scalarized Preferences

Furthermore, there exist an  $\varepsilon \in \mathbb{R}_+$  and a  $\mathbf{t}_0 \in \mathbb{R}^{m-1}$  such that

$$\gamma(h(\mathbf{t}_0 + \varepsilon), \mathbf{u}^{e,k}) > \gamma(h(\mathbf{t}_0), \mathbf{u}^{e,k}) > \gamma(h(\mathbf{t}_0), \mathbf{u}^{e,i}) > \gamma(h(\mathbf{t}_0 + \varepsilon), \mathbf{u}^{e,i}), \quad (3.30)$$

4. Let  $\mathcal{Y}_p = \{\mathbf{u} \in \mathbb{R}_{\geq 0}^2 \mid u_1^2 + u_2^2 = 1\}$ . It follows that for any chosen weights, the minimizers of  $\Psi^{ws}$  are either  $(1, 0)$ ,  $(0, 1)$  or both, which are the extreme points of  $\mathcal{Y}_p$ . Both points are also the minimizers of  $\Psi^t$  on  $\mathcal{Y}_p$ . It follows that  $\Psi^{ws}, \Psi^s, \Psi^t \notin S^{NE}$ .

Let  $\mathcal{Y}'_p = \{(1, 1)\} + \mathcal{Y}$ . It follows that for any chosen weights the minimizers of  $\Psi^{wp}$  are either  $(2, 1)$ ,  $(1, 2)$  or both, which are the extreme points of  $\mathcal{Y}'_p$ . It follows that  $\Psi^*, \Psi^{wp}, \Psi^p \notin S^{NE}$ .

5. The statement  $\Psi^+ \notin S^{NE}$  already follows from  $\Psi^s \notin S^{NE}$ . The following proof shows that not only the weighted sum as special case of the additive scalarization function violates non-extremeness, but that every additive scalarization function violates non-extremeness.

Let  $\Psi^+(\mathbf{u}) = g_1(u_1) + g_2(u_2)$ . Since  $g_1$  and  $g_2$  are strictly increasing, there exist inverse functions  $g_1^{-1}, g_2^{-1}$  such that there exists a  $\mathcal{Y}_p \in \Upsilon_p$  with

$$\mathcal{Y}_p = \{(g_1^{-1}(u_1), g_2^{-1}(u_2)) \in \mathbb{R}^2 \mid \forall u_1, u_2 \geq 0 : u_1^2 + u_2^2 = 1\}. \quad (3.31)$$

It follows that

$$\Psi^+(\mathbf{u}) = g_1(g_1^{-1}(u_1)) + g_2(g_2^{-1}(u_2)) = u_1^2 + u_2^2. \quad (3.32)$$

The minimizers of Equation (3.32) are the extreme points of  $\mathcal{Y}_p$ , since optimizing  $\Psi^+$  on  $\mathcal{Y}_p$  results in finding the minima of the sum of objectives on the unit arc in the positive orthant.

□

The requirements for angle utility to satisfy non-extremeness demand that there exist nonlinear tradeoffs between at least two objectives when converging from the interior of the Pareto front to an extreme point. Nonlinear optimization problems usually satisfy this condition [ZDT00, DTLZ05, HHBW06, ED07]. This requirement, however, is only a sufficient but not necessary condition. The limiting behavior of objectives makes no statement about the angle utility of interior points of the front. The conditions in Proposition 6 merely guarantee that the angle utility of extreme points is larger compared to points in their immediate neighborhood. There may exist interior global scalarization optima independent of the limiting behavior of the different objectives. Even linear tradeoffs allow the existence of interior angle utility optima. The Pareto front

### 3.2. Characterizing Scalarization Functions

of the three objective benchmark problem DTLZ1 [DTLZ05], for example, consists of a symmetric triangle and is thereby linear (Figure 5.7). Its corresponding angle utility optimum is located at the center of the front.

Athan and Papalambros [AP96] proposed two additive scalarization functions for obtaining interior points on concave Pareto fronts: An exponential weighted criteria method that multiplies each objective value by a positive scalar and exponentiates each product. The resulting powers are weighted once more and summed. Weighted compromise programming raises each objective value by a positive scalar. The powers are multiplied by positive weights and subsequently summed. The authors prove that both functions are able to attain every point on any concave Pareto front by changing weights and increasing the positive scalars by which objectives are either multiplied or raised. Thereby, the exponential weighted criteria method and the weighted compromise programming can be configured such that they always obtain non-extreme points. Proposition 6 seemingly contradicts these findings, because all additive scalarization functions are characterized as violating non-extremeness. The difference between Proposition 6 and the results in [AP96] consists of Proposition 6 requiring the weights and scalars to be fixed. Proposition 6 states in this context, that given a fixed set of weights and scalars, it is possible to construct a Pareto front such that the exponential weighted criteria method and weighted compromise programming obtain only extreme points as solutions. Thereby, both statements are reconciled.

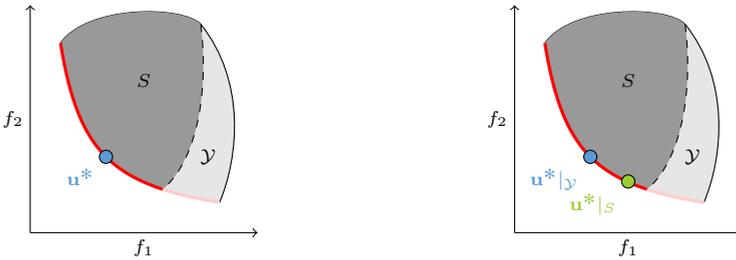
#### 3.2.4. Contraction Consistency

Contraction consistency is a concept from axiomatic bargaining theory. Bargaining theory deals with interactions, in which agents negotiate about the distribution of a surplus that can be achieved if they reach an agreement. If no agreement can be reached, each agent retains her endowment. Such interactions are referred to as bargaining problems. Bargaining theory and social choice theory are closely related, since they aim at deriving socially preferable outcomes from individual preferences. Concepts originating in bargaining theory have been discussed in the context of social choice theory. SWFs in the context of bargaining problems aim at finding allocations – a distribution of the surplus between the agents – that are characterized by desirable properties in the same manner as in social choice theory. Contraction consistency describes the property that if  $x$  is a welfare optimum identified by an SWF on the set of alternatives  $X$ , then  $x$  must also be the welfare optimum on any subset of  $X$  in which  $x$  is contained. In this sense, contraction consistency can be perceived as a weaker form of binary independence [EH89, Roe98].

### 3. Scalarized Preferences

Contraction consistency is a property that directly translates to a characterization of SOOPs. In SOO, if  $x$  is a minimizer of objective function  $f$  on domain  $X$ , then  $x$  is also a minimizer on any subset  $S$  of  $X$  for which  $x \in S$  [Ber99]. Obtaining the optimum to a scalarized MOOP corresponds to solving an SOOP. In contrast to SOO, however, the scalarization value of a point  $\mathbf{u}$  may not only depend on the objective values of  $\mathbf{u}$ , but also on the composition of the feasible set of the objective space  $\mathcal{Y}$ . Thereby, the location of the scalarization optimum may change if the domain of the scalarization function is restricted to a subset of  $\mathcal{Y}$ , even if the original scalarization optimum is still feasible. A scalarization function  $\Psi$  is characterized as contraction consistent if the scalarization optimum  $\mathbf{u}^*$  remains optimal on any restriction of  $\Psi$  for which  $\mathbf{u}^*$  is feasible (see Definition 48 and Figure 3.9).

In real-world applications, contraction consistency is a useful property, since the composition of the feasible set may change after the search for optima has already been completed. In airfoil design, for example, optimal solutions may turn out to be impractical to manufacture [WCL10]. Restrictions in the production technology are difficult to translate into mathematical constraints, which is why such violations are not anticipated in advance. A scalarization function that violates contraction consistency may require restarting the search if the composition of  $\mathcal{Y}$  changes, even if  $\mathbf{u}^*$  remains feasible. Selecting a new optimum even though the old one is available, appears counterintuitive from a decision making perspective.



(a) Satisfaction of contraction consistency.

(b) Violation of contraction consistency.

Figure 3.9.: Illustration of the contraction consistency axiom. If the original feasible set  $\mathcal{Y}$  is reduced to  $S$  ( $S \subset \mathcal{Y}$ ), such that the original scalarization optimum  $\mathbf{u}^*$  of  $\mathcal{Y}$  is still feasible, contraction consistency requires that  $\mathbf{u}^*$  is also the scalarization optimum of  $S$ .

**Definition 48** (contraction consistency cf. [Roe98]). *Let  $\Psi$  be a scalarization function and  $\mathbf{u}^*$  denote the scalarization optimum of  $\Psi$  on  $\mathcal{Y} \in \Upsilon$ . The scalarization function  $\Psi$  fulfills contraction consistency if for all  $S \in \Upsilon$  satisfying  $S \subset \mathcal{Y}$  and  $\mathbf{u}^* \in S$  it*

### 3.2. Characterizing Scalarization Functions

follows that

$$\arg \min_{\mathbf{u} \in S} \Psi(\mathbf{u}) = \mathbf{u}^*. \quad (3.33)$$

The contraction consistency axiom can be used to characterize the robustness of a scalarization function with respect to changes in  $\mathcal{Y}$  (see Proposition 7). Evidently, any scalarization function that violates binary independence remains contraction consistent as long as the additional information obtained from  $\mathcal{Y}$ , which is required for computing the scalarization values, remains the same across different subsets. For example, if the ideal point is identical across multiple subsets of the objective space, the scalarization values computed by the Chebyshev method remain the same. The more information a scalarization function requires about  $\mathcal{Y}$  to remain contraction consistent the less robust it is towards changes in  $\mathcal{Y}$ .

**Proposition 7.** *Let  $S^{CC}$  denote the set of scalarization functions that satisfy contraction consistency. Furthermore, let  $S, \mathcal{Y} \in \Upsilon$  and  $S \subset \mathcal{Y}$ . Then,*

1.  $\Psi^{ws}, \Psi^s, \Psi^{wp}, \Psi^p \in S^{CC}$ ,
2.  $\Psi^{wc}, \Psi^c \in S^{CC}$  for all  $S, \mathcal{Y}$  for which the ideal points coincide,
3.  $\Psi^n \in S^{CC}$  for all  $S, \mathcal{Y}$  for which the nadir points coincide,
4.  $\Psi^a \in S^{CC} \forall S, \mathcal{Y}$  for which the extreme points coincide,
5.  $\Psi^t \in S^{CC}$  for  $m = 2$  and  $\mathcal{Y}_p$  convex or  $\mathcal{Y}_p$  concave and for all  $S, \mathcal{Y}$  for which the extreme points coincide.

*Proof.* Evidently, any scalarization function fulfilling binary independence also satisfies contraction consistency. For any  $i \in \{wc, c, n, a\}$  and all  $\mathbf{u} \in \mathcal{Y}$  it holds that  $\Psi^i|_{\mathcal{Y}}(\mathbf{u}) = \Psi^i|_S(\mathbf{u})$  if either the ideal point, the nadir point or the extreme points coincide, respectively.

Let  $m = 2$  and  $h$  be an implicit function of  $\mathcal{Y}_p$ . If  $\mathcal{Y}_p$  is convex then the tradeoff utility of a point  $\mathbf{u} \in \mathcal{Y}_p$  is equivalent to the maximum of the derivative of  $h$  with respect to  $f_1$  or  $f_2$  at  $\mathbf{u}$ . If  $\Psi^t$  is concave, then either  $\mathbf{u}^{e,1}$  or  $\mathbf{u}^{e,2}$  maximizes the tradeoff utility of  $\mathbf{u}$ . See [SBS13, Theorem 5] for reference.  $\square$

Note that the Nash bargaining solution fully satisfies contraction consistency as it is defined in bargaining theory without any restrictions [Nas50]. This is because the disagreement point, which is equivalent to the nadir point in MOO, is also required to remain feasible. This restriction is considered as special case in Proposition 7.

### 3. Scalarized Preferences

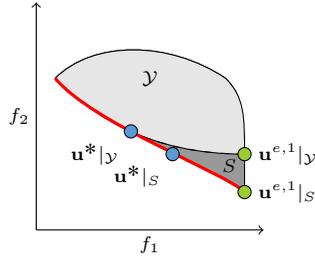


Figure 3.10.: Illustration of the monotonicity axiom. If the ideal value of  $f_1$  decreases (indicated by the downward shift of the extreme point of objective one), then the monotonicity axiom states that the value of  $f_1$  at the scalarization optimum must decrease as well.

#### 3.2.5. Monotonicity

Monotonicity axioms also originate in bargaining theory and take an opposing stance to contraction consistency. There exist differing approaches to formulate monotonicity properties in bargaining theory, however their joint ethical foundations coincide. Whenever the set of alternatives  $X$  increases such that an agent can derive a higher utility from her preferred bargaining outcome, then the welfare optimum of the increased set must not make the agent worse off compared to the optimum of the original set  $X$ . The reasoning behind this axiom is that an agent should not be punished if she is willing or able to bring *more* to the negotiating table [Roe98].

An equivalent logic applies to MOO. In real-world applications, technological advances sometimes allow further improvements in one objective that do not necessarily require the deterioration of other objectives. Technological advances in the automobile industry, for example, have lead to increases in the power output of combustion engines while decreasing fuel consumption at the same time. Such achievements should be rewarded in the decision making process to provide an incentive for actively researching new technologies – or alternative policies in non-engineering scenarios – that improve objective values. Otherwise, such improvements would be ignored in the decision making process or even be detrimental to the corresponding optimal objective value. The monotonicity axiom that is proposed in this work, states that if the ideal value of objective  $i$  becomes smaller while all other ideal values retain their values, the value of  $i$  at the scalarization optimum must decrease as well (see Definition 49).

**Definition 49** (monotonicity cf. [Roe98]). *Let  $\mathbf{u}^*$  denote the scalarization optimum of a scalarization function  $\Psi$  on  $\mathcal{Y} \in \Upsilon$ . The function  $\Psi$  fulfills monotonicity if for all*

### 3.2. Characterizing Scalarization Functions

$S \in \Upsilon$  with  $S \supset \mathcal{Y}$  for which there exists an  $i \in [m]$  such that  $\min_{\mathbf{u} \in S} u_i < \min_{\mathbf{u} \in \mathcal{Y}} u_i$  and for all  $j \in [m] \setminus \{i\}$   $\min_{\mathbf{u} \in S} u_j = \min_{\mathbf{u} \in \mathcal{Y}_p} u_j$  it follows that

$$u_i^* < \left( \arg \min_{\mathbf{u} \in S} \Psi(\mathbf{u}) \right)_i. \quad (3.34)$$

Scalarization functions that satisfy binary independence are obviously indifferent to changes in the ideal values. If a scalarization function violates binary independence or ignores ideal values in computing scalarization values, a decrease in a given ideal value might even lead to a deterioration of the corresponding objective value at the scalarization optimum. In case of the Nash bargaining solution, a decrease of the ideal value that is accompanied by an increase of the nadir value of another objective implies that the objective value of the former at the scalarization optima increases. Even if ideal values are utilized for computing scalarization values, the effect of decreasing an ideal value may be counteracted by other components utilized in computing scalarization values. If extreme points are asymmetric, a small change in an ideal value might translate to a minor change in the corresponding angle that does not affect the maximum of all angles (see Proposition 8). Spending additional effort on improving the individual minima of an MOOP is thereby only beneficial if the chosen scalarization function satisfies monotonicity.

**Proposition 8.** *Let  $S^M$  denote the set of scalarization functions that satisfy monotonicity. Then,*

1.  $\Psi^c, \Psi^{wc} \in S^M$ ,
2.  $\Psi^t \in S^M$  if the following two conditions are met:
  - a) The Pareto front  $\mathcal{Y}_p$  is concave and  $m = 2$ ,
  - b) Given  $\mathcal{Y}, S, i$  and  $j$  according to Definition 49 it must hold that  $u_1^{e,i}|_S + u_2^{e,i}|_S < u_1^{e,j}|_S + u_2^{e,j}|_S$ .
3.  $\Psi^{ws}, \Psi^s, \Psi^{wp}, \Psi^p, \Psi^n, \Psi^a \notin S^M$ .

*Proof.* Let  $\mathbf{u}^*|_{\mathcal{Y}}$  and  $\mathbf{u}^*|_S$  denote the scalarization optima of  $\mathcal{Y}, S \in \Upsilon$ , respectively. Furthermore, let  $i \in [m]$  with  $\min_{\mathbf{u} \in S} u_i < \min_{\mathbf{u} \in \mathcal{Y}} u_i$  and  $\forall j \in [m] \setminus \{i\}$   $\min_{\mathbf{u} \in S} u_j = \min_{\mathbf{u} \in \mathcal{Y}_p} u_j$ .

1. The scalarization optimum of the weighted Chebyshev method is located at the interception of the line  $l$  passing through  $\mathbf{u}^*$  and  $\mathbf{u}^* + (1/\lambda_1, \dots, 1/\lambda_m)$  with the Pareto front. Decreasing  $\mathbf{u}^*$  in objective  $i$  by  $\varepsilon \in \mathbb{R}_+$  leads to a shift of  $l$  in dimension  $i$  by  $\varepsilon$ . It follows that  $u_i^*|_S < u_i^*|_{\mathcal{Y}}$ .

### 3. Scalarized Preferences

2. Let  $\mathbf{u}^{e,i} := \mathbf{u}^{e,i}|_S$  and  $\mathbf{u}^{e,j} := \mathbf{u}^{e,j}|_S$ . If  $m = 2$  and  $\mathcal{Y}_p$  concave then the scalarization optimum coincides with either of the extreme points. If  $\Psi^t(\mathbf{u}^{e,i}) < \Psi^t(\mathbf{u}^{e,j})$  holds then  $\Psi^t$  fulfills monotonicity and it follows that

$$\Psi^t(\mathbf{u}^{e,i}) < \Psi^t(\mathbf{u}^{e,j}) \quad (3.35a)$$

$$\Leftrightarrow \frac{u_i^{e,i} - u_i^{e,j}}{u_j^{e,j} - u_j^{e,i}} < \frac{u_j^{e,j} - u_j^{e,i}}{u_i^{e,i} - u_i^{e,j}} \quad (3.35b)$$

$$\Leftrightarrow (u_i^{e,i} - u_i^{e,j})^2 < (u_j^{e,j} - u_j^{e,i})^2 \quad (3.35c)$$

$$\Leftrightarrow u_i^{e,i} + u_j^{e,i} < u_i^{e,j} + u_j^{e,j} \quad (3.35d)$$

$$\Leftrightarrow u_1^{e,i}|_S + u_2^{e,i}|_S < u_1^{e,j}|_S + u_2^{e,j}|_S \quad (3.35e)$$

3. Evidently, any scalarization function that fulfills binary independence cannot satisfy monotonicity. It follows that  $\Psi^{ws}, \Psi^s, \Psi^{wp}, \Psi^p \notin S^M$ . Only scalarization functions that utilize ideal values in computing scalarization values may satisfy monotonicity. Thereby, the Nash bargaining solution does not satisfy monotonicity, since it only requires knowledge of the nadir point.

For any  $a \in \mathbb{R}_+$  let  $P_a = \{(x, 1 - \sqrt{x}) \in \mathbb{R}^2 \mid x \in [0, a]\}$  with  $\mathcal{Y} = P_1$  and  $S = P_2$ . For angle utility, it follows that  $\mathbf{u}^*|_{\mathcal{Y}} = (0, 0)$ ,  $\mathbf{u}^*|_{\mathcal{Y}} = (0, 1 - \sqrt{2})$  and  $\mathbf{u}^*|_{\mathcal{Y}} \approx (0.3838, 0.3805)$ ,  $\mathbf{u}^*|_S \approx (0.2626, 0.4875)$ . It follows that  $u_2^*|_{\mathcal{Y}} < u_2^*|_S$ , although  $u_1^*|_{\mathcal{Y}} > u_1^*|_S$ .

□

Proposition 8 shows that the majority of the scalarization functions that are discussed within this work, do not satisfy the monotonicity axiom. For angle utility, even an adverse effect can be observed. Decreasing the ideal value of objective  $i$  may lead to an increase of the value of objective  $i$  at the angle utility optimum. Tradeoff utility satisfies monotonicity only under very specific conditions. For two objectives, the conditions state that the Pareto front must be concave and that  $\mathbf{u}^{e,i}|_S$  lies underneath the line with slope  $-1$  that passes through  $\mathbf{u}^{e,j}|_S$ .

#### 3.2.6. Equity

SWFs aim at finding a compromise between the different opinions and needs members of a society possess. Since the well-being of each individual is measured in utility, the social welfare optimum should aim at equally distributing

### 3.2. Characterizing Scalarization Functions

utility among the society. Equity axioms in social choice theory formalize this notion by describing circumstances under which transfers from wealthier to poorer individuals would be socially desirable [dG02]. The Pigou-Dalton principle states that a transfer from a wealthier to a poorer individual is desirable as long as the utility of the poorer individual does not surpass the utility of the wealthier individual after the transfer. Social choice theory distinguishes between a weak and a strong satisfaction of the Pigou-Dalton principle. Weak satisfaction implies that the SWF must be at least indifferent between transfers from the rich to the poor, while strong satisfaction mandates that the outcomes of such transfers must always be preferred to the original allocation [dG02].

In the same manner, scalarization functions aim at striking a balance between the objectives that are optimized. Notions of equity have been adopted in MOO to obtain preferred solutions [SHS10b, KOW04, Eic12, KOW04, Ogr09], however they have not been utilized to explicitly characterize scalarization functions. Evidently, the Pigou-Dalton principle is only applicable if each objective is equally important in the decision making process. Any scalarization function that prioritizes individual objectives is in clear violation of any equity notion. Definition 50 distinguishes between a strong and weak Pigou-Dalton principle in the same manner as in social choice theory. The strict variant mandates that decreasing a large objective while increasing a small objective by the same amount must always be preferred, while the weak variant requires the scalarization function to be at least indifferent. See Figure 3.11 for an illustration of the concept.

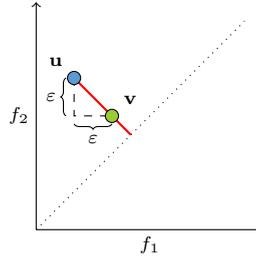


Figure 3.11.: Illustration of the Pigou-Dalton principle. A scalarization function that satisfies the strong Pigou-Dalton principle must prefer any point on the red line to  $\mathbf{u}$ .

**Definition 50** (Pigou-Dalton principle cf. [dG02]). *Let for all  $i, j \in [m]$ , for all  $\mathbf{u}, \mathbf{v} \in \mathcal{Y} \in \Upsilon$  and for all  $\varepsilon \in \mathbb{R}_+$  the following two conditions hold:*

- $v_j < v_i, v_j = u_j + \varepsilon, v_i = u_i - \varepsilon$  and
- $\forall k \notin \{i, j\}, v_k = u_k$ .

### 3. Scalarized Preferences

A scalarization function  $\Psi$  fulfills

- the weak Pigou-Dalton principle if  $\Psi(\mathbf{u}) \geq \Psi(\mathbf{v})$  and
- the strong Pigou-Dalton principle if  $\Psi(\mathbf{u}) > \Psi(\mathbf{u})$ .

Proposition 9 shows which scalarization functions satisfy the Pigou-Dalton principle. Evidently, any scalarization function that satisfies the strong Pigou-Dalton principle also satisfies the weak Pigou-Dalton principle. In addition, a function that satisfies the weak but not the strong Pigou-Dalton principle is indifferent towards Pigou-Dalton transfers as described in Definition 50.

**Proposition 9.** *Let  $S^{WPDP}$  and  $S^{SPDP}$  denote the set of scalarization functions that satisfy the weak and strong Pigou-Dalton principle, respectively. Then,*

1.  $\Psi^p \notin S^{SPDP}$  and  $\Psi^p \notin S^{WPDP}$ ,
2.  $\Psi^c \in S^{SPDP}$  for  $m = 2$  and  $\Psi^c \in S^{WPDP}$  for  $m > 2$ , and additionally for all  $m$  it must hold that for all  $i, j \in [m] : u_i^* = u_j^*$ ,
3.  $\Psi^n \in S^{SPDP}$  if for all  $i, j \in [m] : u_i^{ndr} = u_j^{ndr}$ ,
4.  $\Psi^a \in S^{WPDP}$  if  $m = 2$  and the extreme points are symmetric; additionally  $\Psi^a \in S^{SPDP}$  if  $\mathbf{u} \notin \{\mathbf{w} \in \mathbb{R}^2 \mid \exists t \in [0, 1] : \mathbf{w} = \mathbf{u}^{e,1} + t(\mathbf{u}^{e,2} - \mathbf{u}^{e,1})\}$ , i.e.  $\mathbf{u}$  does not lie on the line that connects both extreme points,
5.  $\Psi^s \in S^{WPDP}$ ,
6.  $\Psi^t \notin S^{WPDP}$ .

*Proof.* Let  $i, j \in [m]$ ,  $\varepsilon \in \mathbb{R}_+$ ,  $\mathbf{u}, \mathbf{v} \in \mathcal{Y} \in \Upsilon$  such that  $v_j < v_i$ ,  $v_j = u_j + \varepsilon$ ,  $v_i = u_i - \varepsilon$  and for all  $l \notin \{i, j\}$  let  $v_l = u_l$ .

1. For  $\Psi^p$  it holds that

$$\Psi^p(\mathbf{u}) = u_i \cdot u_j \cdot \prod_{k \in [m] \setminus \{i, j\}} u_k < v_i v_j \prod_{k \in [m] \setminus \{i, j\}} u_k = \Psi^p(\mathbf{v}) \quad (3.36a)$$

$$\Leftrightarrow (v_i + \varepsilon)(v_j - \varepsilon) < v_i v_j \quad (3.36b)$$

$$\Leftrightarrow v_i v_j - v_i \varepsilon + v_j \varepsilon - \varepsilon^2 < v_i v_j \quad (3.36c)$$

$$\Leftrightarrow -v_i + v_j - \varepsilon < 0 \quad (3.36d)$$

$$\Leftrightarrow v_j < v_i + \varepsilon. \quad (3.36e)$$

Equation (3.36e) is true, because  $v_j < v_i$  and  $\varepsilon > 0$ .

### 3.2. Characterizing Scalarization Functions

2. For  $\Psi^c$  and  $m = 2$  it holds that

$$\Psi^c(\mathbf{u}) = \max(u_i - u_i^*, u_j - u_j^*) = \max(v_i + \varepsilon - u_i^*, v_j - \varepsilon - u_j^*), \quad (3.37a)$$

$$\Psi^c(\mathbf{v}) = \max(v_i - u_i^*, v_j - u_j^*), \quad (3.37b)$$

$$\Rightarrow v_i + \varepsilon - u_i^* > v_i - u_i^* > v_j - v_j^* > v_j - \varepsilon - u_j^*. \quad (3.37c)$$

Equation (3.37c) is true, since we require that  $u_i^* = u_j^*$ . If  $m > 2$ , then  $\Psi^c(\mathbf{u}) = \Psi^c(\mathbf{v})$  if  $i \neq \arg \max_{k \in [m]} u_i$ .

3. For  $\Psi^n$  it holds that

$$\begin{aligned} \Psi^p(\mathbf{u}) &= -(u_i^{ndr} - u_i)(u_j^{ndr} - u_j) \cdot \prod_{k \in [m] \setminus \{i, j\}} (u_k^{ndr} - u_k) > \\ &\quad -(u_i^{ndr} - v_i)(u_j^{ndr} - v_j) \prod_{k \in [m] \setminus \{i, j\}} (u_k^{ndr} - u_k) = \Psi^p(\mathbf{v}) \end{aligned} \quad (3.38a)$$

$$\Leftrightarrow (u_i^{ndr} - u_i)(u_j^{ndr} - u_j) < (u_i^{ndr} - v_i)(u_j^{ndr} - v_j) \quad (3.38b)$$

$$\Leftrightarrow (u_i^{ndr} - (v_i + \varepsilon))(u_j^{ndr} - (v_j - \varepsilon)) < (u_i^{ndr} - v_i)(u_j^{ndr} - v_j) \quad (3.38c)$$

$$\begin{aligned} \Leftrightarrow u_i^{ndr} u_j^{ndr} - u_i^{ndr} v_j + u_i^{ndr} \varepsilon - v_i u_j^{ndr} + v_i v_j - v_i \varepsilon - \varepsilon u_j^{ndr} + \\ \varepsilon v_j - \varepsilon^2 < u_i^{ndr} u_j^{ndr} - u_i^{ndr} v_j - v_i u_j^{ndr} + v_i v_j \end{aligned} \quad (3.38d)$$

$$\Leftrightarrow u_i^{ndr} - v_i - u_j^{ndr} + v_j - \varepsilon < 0 \quad (3.38e)$$

$$\Leftrightarrow v_j - u_j^{ndr} < v_i - u_i^{ndr} + \varepsilon. \quad (3.38f)$$

Equation (3.38f) is true, since we require that  $u_i^{ndr} = u_j^{ndr}$ .

4. For  $\Psi^a$  and  $m = 2$  let us assume without loss of generality that  $\mathbf{u}^{e,1} = (1, 0)$  and  $\mathbf{u}^{e,2} = (0, 1)$ ,<sup>3</sup>  $i = 1, j = 2$ . It follows that for any  $\mathbf{u} \in \mathcal{Y}_p$

$$\gamma(\mathbf{u}, \mathbf{u}^{e,1}) = \arctan\left(\frac{u_2}{1 - u_1}\right) = \arctan\left(\frac{v_2 - \varepsilon}{1 - v_1 - \varepsilon}\right), \quad (3.39a)$$

$$\gamma(\mathbf{u}, \mathbf{u}^{e,2}) = \arctan\left(\frac{u_1}{1 - u_2}\right) = \arctan\left(\frac{v_1 + \varepsilon}{1 - v_2 + \varepsilon}\right), \quad (3.39b)$$

$$\gamma(\mathbf{v}, \mathbf{u}^{e,1}) = \arctan\left(\frac{v_2}{1 - v_1}\right), \quad (3.39c)$$

$$\gamma(\mathbf{v}, \mathbf{u}^{e,2}) = \arctan\left(\frac{v_1}{1 - v_2}\right). \quad (3.39d)$$

Three cases can be distinguished:

- a) If  $\gamma(\mathbf{v}, \mathbf{u}^{e,1}) < \gamma(\mathbf{v}, \mathbf{u}^{e,2})$  then  $v_1 \in (v_2, 1 - v_2)$ . It follows that  $\Psi^a$  satisfies the strong Pigou-Dalton principle if  $\gamma(\mathbf{v}, \mathbf{u}^{e,2}) < \gamma(\mathbf{u}, \mathbf{u}^{e,2})$ , which is equivalent to  $v_1 < 1 - v_2$ .

<sup>3</sup>See Proposition 10 for a justification of this simplification.

### 3. Scalarized Preferences

- b) If  $\gamma(\mathbf{v}, \mathbf{u}^{e,1}) > \gamma(\mathbf{v}, \mathbf{u}^{e,2})$  then either  $v_1 > 1 - v_2$  or  $v_1 < v_2$ , whereas  $v_1 < v_2$  violates the condition  $v_j < v_i$ . It follows that  $\Psi^a$  satisfies the strong Pigou-Dalton principle if  $\gamma(\mathbf{v}, \mathbf{u}^{e,1}) < \gamma(\mathbf{u}, \mathbf{u}^{e,1})$ :

$$\gamma(\mathbf{v}, \mathbf{u}^{e,1}) < \gamma(\mathbf{u}, \mathbf{u}^{e,1}) \quad (3.40a)$$

$$\Leftrightarrow \frac{v_2}{1 - v_1} < \frac{v_2 - \varepsilon}{1 - v_1 - \varepsilon} \quad (3.40b)$$

$$\Leftrightarrow v_2(1 - v_1 - \varepsilon) < (v_2 - \varepsilon)(1 - v_1) \quad (3.40c)$$

$$\Leftrightarrow v_2 - v_2v_1 - v_2\varepsilon < v_2 - v_2v_1 - \varepsilon + \varepsilon v_1 \quad (3.40d)$$

$$\Leftrightarrow -v_2 < -1 + v_1 \quad (3.40e)$$

$$\Leftrightarrow v_1 > 1 - v_2. \quad (3.40f)$$

- c) If  $\gamma(\mathbf{v}, \mathbf{u}^{e,1}) = \gamma(\mathbf{v}, \mathbf{u}^{e,2})$  then  $v_1 = 1 - v_2$ . (The case  $v_1 = v_2$  would violate the condition  $v_j < v_i$ .) It follows that

$$\gamma(\mathbf{v}, \mathbf{u}^{e,1}) = \gamma(\mathbf{u}, \mathbf{u}^{e,1}) = \gamma(\mathbf{v}, \mathbf{u}^{e,2}) = \gamma(\mathbf{u}, \mathbf{u}^{e,2}) = \pi/4. \quad (3.41)$$

All angles possessing the value  $\pi/4$  ( $45^\circ$  in degrees) implies that  $\mathbf{u}$  and  $\mathbf{v}$  are located on the straight line  $L$  that connects both extreme points, i.e.  $\mathbf{u}, \mathbf{v} \in L$  with  $L = \{\mathbf{w} \in \mathbb{R}^2 \mid \exists t \in [0, 1] : \mathbf{w} = \mathbf{u}^{e,1} + t(\mathbf{u}^{e,2} - \mathbf{u}^{e,1})\}$ .

In cases one and two, it follows that  $\Psi^a \in S^{SPDP}$ . Case three implies that  $\Psi^a \in S^{WPDP}$ .

5. For  $\Psi^s$  it holds that

$$\Psi^s(\mathbf{u}) = u_i + u_j + \sum_{k \in [m] \setminus \{i,j\}} u_k \geq v_i + v_j + \sum_{k \in [m] \setminus \{i,j\}} u_k = \Psi^s(\mathbf{v}) \quad (3.42a)$$

$$\Leftrightarrow u_i + u_j \geq v_i + v_j \quad (3.42b)$$

$$\Leftrightarrow v_i + \varepsilon + v_j - \varepsilon \geq v_i + v_j. \quad (3.42c)$$

Since Equation (3.42c) is not strict,  $\Psi^s$  satisfies the weak, but not the strong Pigou-Dalton principle.

6. Let  $\mathcal{Y}_p = \{\mathbf{u}, \mathbf{v}, \mathbf{w}\}$  with  $\mathbf{u} = (10, 0)$ ,  $\mathbf{v} = (5, 5)$  and  $\mathbf{w} = (6, 0)$ . It follows that  $\Psi^t(\mathbf{u}) < \Psi^t(\mathbf{v})$ , although  $\mathbf{v}$  should be weakly preferred to  $\mathbf{u}$  according to the weak Pigou-Dalton principle. It follows that  $\Psi^t \notin S^{WPDP}$ .

□

### 3.2. Characterizing Scalarization Functions

Reference points play a crucial role in the satisfaction of the equity axiom for scalarization functions that depend on such points. The Chebyshev method requires all components of the nadir point to possess the same value. Otherwise, transfers that are desirable in the Pigou-Dalton sense might even increase the scalarization value. Consider the example  $\mathbf{u}^* = (5, 0)$ ,  $\mathbf{u} = (10, 5)$  and  $\mathbf{v} = (7.5, 7.5)$ . The Pigou-Dalton principle dictates that  $\mathbf{v}$  should be weakly preferred to  $\mathbf{u}$ , whereas the Chebyshev method prefers  $\mathbf{u}$  to  $\mathbf{v}$ , since  $\Psi^c(\mathbf{u}) = 5 < 7.5 = \Psi^c(\mathbf{v})$ . From the proof of Proposition 9 one can argue, that the Chebyshev method deems Pigou-Dalton transfers desirable if the distance to ideal values is factored in as well. For more than two objectives, the Chebyshev method is indifferent towards transfers that do not lead to a reduction of the objective value that is furthest away from its ideal value. The Nash bargaining solution factors in the distances of objectives to their nadir values. Equation (3.38a) shows that the Pigou-Dalton transfer must compensate for the difference in the nadir values of the corresponding objectives involved in the transfer. Extreme points are required to be symmetric for angle utility. Angle utility satisfies the strong Pigou-Dalton principle, as long as the Pareto front is not a straight line that connects both extreme points. This observation offers an explanation to why angle utility has a strong tendency to favor interior points on both convex and concave Pareto fronts. If the Pareto front consists of a straight line between both extreme points, angle utility is indifferent towards Pigou-Dalton transfers. Scalarization functions that violate the Pigou-Dalton principle deem Pigou-Dalton transfers harmful and are less likely to identify a scalarization optimum that provides a balance between the different objectives.

Evidently, the Pigou-Dalton principle requires that objective values are measured on a common scale such that they are comparable. Otherwise, a transfer by a fixed amount from a larger to a smaller objective may imply a different usefulness depending on the chosen scales of the given objectives. For example, increasing the power output of a car engine by 1 kW at the cost of consuming one more liter of gasoline is less desirable than decreasing the gas consumption by one liter while reducing the power output by 1 kW. The effect of different scales for measuring objective values is therefore analyzed next.

#### 3.2.7. Invariance to Scaling

The utility function of a single individual is a representation of her preferences on the set of available alternatives  $X$ . The numeric values express preferences by assigning a larger utility to more desirable options. Any strictly increasing transformation of the utility function results in the same individual ranking on  $X$ . It is therefore desirable that the social ranking induced by an SWF on

### 3. Scalarized Preferences

$X$  is invariant to transformations of the individual utility functions as long as individual rankings on  $X$  remain the same [dG02].

In economics, there exist ordinal and cardinal utility functions (cf. Section 1.2). Cardinal utility functions allow statements about preference intensity between different alternatives. The larger the difference in utility between two options, the stronger is the preference for the alternative possessing the larger utility. Differences of ordinal utility, on the other hand, allow now statement about the preference intensity. Thereby, ordinal utility functions are unique up to strictly increasing transformations and cardinal utility functions are unique up to affine transformations [VNM53].

Social choice theory categorizes the invariance of SWFs with respect to different types of transformations of utility functions. These transformations include increasing and affine transformations depending on whether utility is measured on an ordinal or cardinal scale. Furthermore, the utility functions of the members of the society may either be transformed individually or uniformly [Roe98].

A single objective may be measured on different scales in MOO. In engineering applications for example, there exist multiple units to measure the same physical quantities. Temperature is commonly measured in Celsius, Kelvin or Fahrenheit. Costs and profits may be quantified in different currencies in business or economics problems. The chosen scale of the given objective should not affect its value at the scalarization optimum. Scale transformations can be decomposed into multiple affine transformations. Definition 51 formalizes the notion of scale invariance and introduces multiple affine transformations that can be considered in the context of scalarization in MOO.

**Definition 51** (scale invariance cf. [dG02, Roe98]). *Let  $\psi : \mathbb{R}^m \rightarrow \mathbb{R}^m$  be an affine transformation function. A scalarization function  $\Psi$  is scale invariant with respect to  $\psi$  if for all  $\mathcal{Y} \in \Upsilon$ ,  $\mathbf{u}, \mathbf{v} \in \mathcal{Y}$  and all  $R \in \{<, =, >\}$  it holds that*

$$\Psi|_{\mathcal{Y}}(\mathbf{u})R\Psi|_{\mathcal{Y}}(\mathbf{v}) \Rightarrow \Psi|_{\psi(\mathcal{Y})}(\psi(\mathbf{u}))R\Psi|_{\psi(\mathcal{Y})}(\psi(\mathbf{v})). \quad (3.43)$$

Let  $a \in \mathbb{R}_+$ ,  $b \in \mathbb{R}$  and for all  $i \in [m]$  let  $a_i \in \mathbb{R}_+$  and  $b_i \in \mathbb{R}$ . For any  $\mathbf{u} \in \mathcal{Y}$  the following affine transformations are distinguished:

1.  $[a_i, b_i]$ :  $\psi(\mathbf{u}) = (a_1u_1 + b_1, \dots, a_mu_m + b_m)$ ,
2.  $[a_i, b]$ :  $\psi(\mathbf{u}) = (a_1u_1 + b, \dots, a_mu_m + b)$ ,
3.  $[a, b_i]$ :  $\psi(\mathbf{u}) = (au_1 + b_1, \dots, au_m + b_m)$ ,
4.  $[a_i, 0]$ :  $\psi(\mathbf{u}) = (a_1u_1, \dots, a_mu_m)$ ,
5.  $[1, b_i]$ :  $\psi(\mathbf{u}) = (u_1 + b_1, \dots, u_m + b_m)$ ,

### 3.2. Characterizing Scalarization Functions

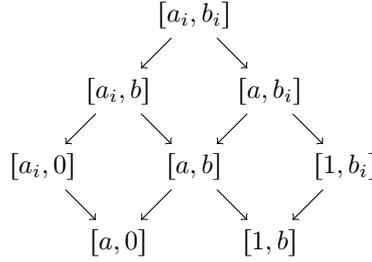


Figure 3.12.: A diagram showing the relationships between the different affine transformations of Definition 51. An arrow pointing from an affine transformation  $\psi_1$  to an affine transformation  $\psi_2$  implies that any scalarization function  $\Psi$  that is scale invariant with respect to  $\psi_1$  is also scale invariant with respect to  $\psi_2$ . Conversely, if  $\Psi$  is not scale invariant with respect to  $\psi_2$ , then  $\Psi$  is also not scale invariant with respect to  $\psi_1$ .

6.  $[a, 0]$ :  $\psi(\mathbf{u}) = (au_1, \dots, au_m)$ ,
7.  $[1, b]$ :  $\psi(\mathbf{u}) = (u_1 + b, \dots, u_m + b)$ .

The affine transformations of Definition 51 can be arranged in a hierarchical structure as illustrated in Figure 3.12. Transformations at higher positions in the hierarchy comprise transformations at lower positions. Affine transformations such as  $[a_i, b_i]$  can be decomposed into a change of the *magnitude* ( $a_i$ ) and the *origin* ( $b_i$ ). Magnitude and origin transformations can either be common for all objectives (e.g.  $[a, b]$ ) or be different for each individual objective (e.g.  $[a_i, b_i]$ ). Physical units that measure the same quantity often share a common origin but possess different magnitudes.<sup>4</sup> Time, for example, can be measured in seconds and minutes. The duration of zero seconds is equivalent to zero minutes. Magnitudes, however, usually differ across physical units. Minutes are translated to seconds by the affine transformation  $[60, 0]$ . The position of a scalarization optimum may therefore change, in case the physical unit of a single objective is changed in the problem formulation, if the scalarization function is not invariant with respect to  $[a_i, 0]$ -transformations. This implies that already the choice of the scale by which objectives are measured may affect the optimization outcome if the scalarization methodology is applied.

**Proposition 10.** *Let  $S^\psi$  be the set of scalarization functions that are scale invariant with respect to the affine transformation  $\psi$ . Then,*

<sup>4</sup>There exist, however, also physical units that do not share a common origin. Zero degree Kelvin, for example, are not equivalent to zero degree Celsius.

### 3. Scalarized Preferences

1.  $\Psi^{ws}, \Psi^s, \Psi^{wc}, \Psi^c, \Psi^a, \Psi^t \in S^{[a, b_i]}$  and  $\Psi^{ws}, \Psi^s, \Psi^{wc}, \Psi^c, \Psi^a, \Psi^t \notin \Psi^{[a_i, 0]}$ ,
2.  $\Psi^{wp}, \Psi^p \in S^{[a_i, 0]}$  and  $\Psi^{wp}, \Psi^p \notin S^{[1, b]}$ ,
3.  $\Psi^n \in S^{[a_i, b_i]}$ .

*Proof.* Let  $a \in \mathbb{R}_+, b \in \mathbb{R}$  and for all  $i \in [m]$  let  $a_i \in \mathbb{R}_+$  and  $b_i \in \mathbb{R}$ . For any  $\mathcal{Y} \in \Upsilon$  and  $\mathbf{u}, \mathbf{v} \in \mathcal{Y}$  without loss of generality, let  $\Psi|_{\mathcal{Y}}(\mathbf{u}) < \Psi|_{\mathcal{Y}}(\mathbf{v})$ . Then:

- For  $\Psi^{ws}$  and  $\psi = [a, b_i]$  it holds that

$$\Psi^{ws}|_{\psi(\mathcal{Y})}(\psi(\mathbf{u})) < \Psi^{ws}|_{\psi(\mathcal{Y})}(\psi(\mathbf{v})) \quad (3.44a)$$

$$\Leftrightarrow \sum_{i \in [m]} \lambda_i (au_i + b_i) < \sum_{i \in [m]} \lambda_i (av_i + b_i) \quad (3.44b)$$

$$\Leftrightarrow a \left( \sum_{i \in [m]} \lambda_i u_i \right) + \sum_{i \in [m]} \lambda_i b_i < a \left( \sum_{i \in [m]} \lambda_i v_i \right) + \sum_{i \in [m]} \lambda_i b_i \quad (3.44c)$$

$$\Leftrightarrow \sum_{i \in [m]} \lambda_i u_i < \sum_{i \in [m]} \lambda_i v_i \quad (3.44d)$$

$$\Leftrightarrow \Psi^{ws}|_{\mathcal{Y}}(\mathbf{u}) < \Psi^{ws}|_{\mathcal{Y}}(\mathbf{v}). \quad (3.44e)$$

It follows that  $\Psi^{ws}, \Psi^s \in S^{[a, b_i]}$ .

- For  $\Psi^{wc}$  and  $\psi = [a, b_i]$  it holds that

$$\Psi^{wc}|_{\psi(\mathcal{Y})}(\psi(\mathbf{u})) < \Psi^{wc}|_{\psi(\mathcal{Y})}(\psi(\mathbf{v})) \quad (3.45a)$$

$$\Leftrightarrow \max_{i \in [m]} (\lambda_i (au_i + b_i - (au_i^* + b_i))) < \max_{i \in [m]} (\lambda_i (av_i + b_i - (av_i^* + b_i))) \quad (3.45b)$$

$$\Leftrightarrow a \max_{i \in [m]} (\lambda_i (u_i - u_i^*)) < a \max_{i \in [m]} (\lambda_i (v_i - v_i^*)) \quad (3.45c)$$

$$\Leftrightarrow \max_{i \in [m]} (\lambda_i (u_i - u_i^*)) < \max_{i \in [m]} (\lambda_i (v_i - v_i^*)) \quad (3.45d)$$

$$\Leftrightarrow \Psi^{wc}|_{\mathcal{Y}}(\mathbf{u}) < \Psi^{wc}|_{\mathcal{Y}}(\mathbf{v}). \quad (3.45e)$$

It follows that  $\Psi^{wc}, \Psi^c \in S^{[a, b_i]}$ .

- For  $\Psi^a$  and  $\psi = [a, b_i]$  it holds for any  $k \in [m]$  that

$$\gamma(\psi(\mathbf{u}), \psi(\mathbf{u}^{e,k})) = \arctan \left( \frac{\sqrt{\sum_{i=1, i \neq k}^m (au_i + b_i - (au_i^{e,k} - b_i))^2}}{|au_k - b_i - (au_k^{e,k} - b_i)|} \right) \quad (3.46a)$$

### 3.2. Characterizing Scalarization Functions

$$= \arctan \left( \frac{a \sqrt{\sum_{i=1, i \neq k}^m (u_i - u_i^{e,k})^2}}{a |u_k - u_k^{e,k}|} \right) \quad (3.46b)$$

$$= \arctan \left( \frac{\sqrt{\sum_{i=1, i \neq k}^m (u_i - u_i^{e,k})^2}}{|u_k - u_k^{e,k}|} \right) \quad (3.46c)$$

$$= \gamma(\mathbf{u}, \mathbf{u}^{e,k}), \quad (3.46d)$$

which implies that  $\Psi^a|_{\mathcal{Y}}(\mathbf{u}) = \Psi^a|_{\psi(\mathcal{Y})}(\psi(\mathbf{u}))$ . It follows that  $\Psi^a \in S^{[a, b_i]}$ .

- For  $\Psi^t$  and  $\psi = [a, b_i]$  it holds that

$$\Psi^t|_{\psi(\mathcal{Y})}(\psi(\mathbf{u})) = \max_{\psi(\mathbf{v}) \in \psi(\mathcal{Y})} \frac{\max_{i \in [m]} \psi(u_i) - \psi(v_i)}{\max_{i \in [m]} \psi(v_i) - \psi(u_i)} \quad (3.47a)$$

$$= \max_{\mathbf{v} \in \mathcal{Y}} \frac{\max_{i \in [m]} (au_i - b_i) - (av_i - b_i)}{\max_{i \in [m]} (av_i - b_i) - (au_i - b_i)} \quad (3.47b)$$

$$= \max_{\mathbf{v} \in \mathcal{Y}} \frac{a \max_{i \in [m]} u_i - v_i}{a \max_{i \in [m]} v_i - u_i} \quad (3.47c)$$

$$= \max_{\mathbf{v} \in \mathcal{Y}} \frac{\max_{i \in [m]} u_i - v_i}{\max_{i \in [m]} v_i - u_i} \quad (3.47d)$$

$$= \Psi^t|_{\mathcal{Y}}(\mathbf{u}). \quad (3.47e)$$

It follows that  $\Psi^t \in S^{[a, b_i]}$ .

- Let  $\mathcal{Y} = \{\mathbf{u}, \mathbf{v}\}$  with  $\mathbf{u} = (1, 0)$ ,  $\mathbf{v} = (0, 2)$ ,  $(a_1, a_2) = (3, 1)$  and  $\psi = [a_i, 0]$ . Additionally, let  $\lambda = (1, 1)$ ,  $\mathbf{u}^{e,1} = (3, 0)$  and  $\mathbf{u}^{e,2} = (0, 3)$ . It follows that for all  $i \in \{ws, s, wc, c, a, t\}$

$$\Psi^i|_{\mathcal{Y}}(\mathbf{u}) < \Psi^i|_{\mathcal{Y}}(\mathbf{v}), \quad (3.48)$$

however at the same time

$$\Psi^i|_{\psi(\mathcal{Y})}(\psi(\mathbf{v})) < \Psi^i|_{\psi(\mathcal{Y})}(\psi(\mathbf{u})), \quad (3.49)$$

which implies  $\Psi^i \notin S^{[a_i, 0]}$ .

- For  $\Psi^{wp}$  and  $\psi = [a_i, 0]$  it holds that

$$\Psi^{wp}|_{\psi(\mathcal{Y})}(\psi(\mathbf{u})) < \Psi^{wp}|_{\psi(\mathcal{Y})}(\psi(\mathbf{v})) \quad (3.50a)$$

$$\Leftrightarrow \prod_{i \in [m]} (a_i u_i)^{\lambda_i} < \prod_{i \in [m]} (a_i v_i)^{\lambda_i} \quad (3.50b)$$

### 3. Scalarized Preferences

$$\Leftrightarrow \prod_{i \in [m]} a_i^{\lambda_i} \prod_{i \in [m]} u_i^{\lambda_i} < \prod_{i \in [m]} a_i^{\lambda_i} \prod_{i \in [m]} v_i^{\lambda_i} \quad (3.50c)$$

$$\Leftrightarrow \prod_{i \in [m]} u_i^{\lambda_i} < \prod_{i \in [m]} v_i^{\lambda_i} \quad (3.50d)$$

$$\Leftrightarrow \Psi^{wp}|_{\mathcal{Y}}(\mathbf{u}) < \Psi^{wp}|_{\mathcal{Y}}(\mathbf{v}) \quad (3.50e)$$

It follows that  $\Psi^{wp}, \Psi^p \in S^{[a_i, 0]}$ .

Let  $\mathbf{u} = (10, 1)$ ,  $\mathbf{v} = (5, 5)$ ,  $\boldsymbol{\lambda} = (1, 1)$ ,  $b = 20$  and  $\psi = [1, b]$ . It follows that  $\Psi^{wp}(\mathbf{u}) < \Psi^{wp}(\mathbf{v})$ , however  $\Psi^{wp}(\mathbf{u} + b) < \Psi^{wp}(\mathbf{u} + b)$ , which implies  $\Psi^{wp}, \Psi^p \notin S^{[1, b]}$ .

- For  $\Psi^n$  and  $\psi = [a_i, b_i]$  it holds that

$$\Psi^n|_{\psi(\mathcal{Y})}(\psi(\mathbf{u})) < \Psi^n|_{\psi(\mathcal{Y})}(\psi(\mathbf{v})) \quad (3.51a)$$

$$\Leftrightarrow - \prod_{i \in [m]} ((a_i u_i^{ndr} - b_i) - (a_i u_i - b_i)) < - \prod_{i \in [m]} ((a_i u_i^{ndr} - b_i) - (a_i v_i - b_i)) \quad (3.51b)$$

$$\Leftrightarrow - \prod_{i \in [m]} a_i (u_i^{ndr} - u_i) < - \prod_{i \in [m]} a_i (u_i^{ndr} - v_i) \quad (3.51c)$$

$$\Leftrightarrow - \prod_{i \in [m]} (u_i^{ndr} - u_i) < - \prod_{i \in [m]} (u_i^{ndr} - v_i) \quad (3.51d)$$

$$\Leftrightarrow \Psi^n|_{\mathcal{Y}}(\mathbf{u}) < \Psi^n|_{\mathcal{Y}}(\mathbf{v}) \quad (3.51e)$$

It follows that  $\Psi^n \in S^{[a_i, b_i]}$ .

□

Proposition 10 shows that the majority of the scalarization functions presented in this chapter are not scale invariant with respect to individual changes in magnitudes ( $[a_i, 0]$ -transformations). A common approach to address this issue is normalization [GEF16, Deb01, CCLVV07]. Normalization scales all Pareto optimal points to the range  $[0, 1]$  (see Definition 52).

**Definition 52** (normalization cf. [MA04]). *Normalization is an affine transformation  $\psi$  of type  $[a_i, b_i]$  such that for any  $\mathbf{u} \in \mathcal{Y}$  and  $i \in [m]$*

$$\psi(\mathbf{u})_i = \frac{u_i - u_i^*}{u_i^{ndr} - u_i^*}. \quad (3.52)$$

Although this technique makes objectives measured on different scales comparable, it makes the location of the scalarization optimum dependent on the

### 3.2. Characterizing Scalarization Functions

composition of the Pareto front even if the scalarization function satisfies binary independence. The ideal and nadir point need to be computed even if they are not used in the calculation of scalarization values. In case nadir or ideal values take on different values due to additional constraints being imposed on the MOOP, the location of the scalarization optimum changes as well.

Table 3.2 summarizes the results of this section. The axioms that are proposed in this chapter can also be used to characterize other scalarization functions that have not been included in this analysis – see [MA04] for example for a comprehensive overview of existing scalarization methodologies in MOO. Categorizing scalarization functions according to which axioms they fulfill enables a DM to choose from an entire set of methods that reflects her preferences best if she deems specific axioms more important than others. The proposed axioms can also be used as guidelines to design new scalarization functions. An expert can specifically develop a scalarization function that satisfies a set of preferred axioms or possibly modify an existing scalarization function such that it satisfies these axioms. Normalization, for example, can be combined with any scalarization function presented in Section 3.1 to create a function that is invariant with respect to  $[a_i, b_i]$  transformations.

### 3. Scalarized Preferences

Table 3.2.: Summary of how the presented scalarization functions satisfy the proposed axioms. A checkmark ✓ indicates that the scalarization function completely satisfies the axiom, whereas a checkmark in parentheses (✓) states that the function satisfies the axiom if a set of well-defined restrictions is met that can be verified with reasonable effort by the DM. The cross ✗ states that the scalarization function violates the axiom.

	Pareto compliance	Binary independence	Non-extremeness	Contraction consistency	Monotonicity	Strong Pigou-Dalton	Weak Pigou-Dalton	Scale invariance
Additive scalarization function	✓		✗					
Weighted sum	✓	✓	✗	✓	✗			$[a, b_i]$
Sum of objectives	✓	✓	✗	✓	✗	✗	✓	$[a, b_i]$
Multiplicative scalarization function	(✓)		✗					
Weighted product	(✓)	✓	✗	✓	✗			$[a_i, 0]$
Product of objectives	(✓)	✓	✗	✓	✗	✗	✗	$[a_i, 0]$
Maximal scalarization function	✗							
Weighted Chebyhsev method	✗	✗	✓	(✓)	✓			$[a, b_i]$
Chebyshev method	✗	✗	✓	(✓)	✓	✗	(✓)	$[a, b_i]$
Nash bargaining solution	(✓)	✗	✓	(✓)	✗	(✓)	(✓)	$[a_i, b_i]$
Angle utility	(✓)	✗	(✓)	(✓)	✗	(✓)	(✓)	$[a, b_i]$
Tradeoff utility	✗	✗	✗	(✓)	✗	✗	✗	$[a, b_i]$

## 4. Methodologies for Solving Multi-objective Optimization Problems

The solution to a real-valued MOOP as of Definition 9 is a set of points that potentially consists of an infinite number of elements. Furthermore, calculating objective values for real-world optimization problems often requires a large amount of computational resources. Additionally, a DM is usually only able to implement a single solution of the entire Pareto optimal set. In practical applications, it is therefore often neither tractable nor desirable to obtain the entire set of Pareto optimal solutions. Instead, different methodologies and paradigms for solving MOOPs have been developed to compute only relevant subsets of the entire Pareto optimal set. These techniques are often tailored to or guided by a DM's preference. Categorizing these techniques and paradigms results in an algorithmic characterization of preferences.

This chapter commences by a brief discussion of existing frameworks for algorithmically characterizing preferences in MOO. Subsequently, a new categorization is presented that characterizes solving paradigms based on the optimization outcome – the search result that is generated by an optimization algorithm. A special emphasis is put on analyzing these paradigms from the perspective of scalarized preferences. Two paradigms are proposed that constitute new approaches to characterize optimization outcomes. The remainder of the chapter focuses on algorithms for solving MOOPs. A brief overview of existing methodologies is given and the class of evolutionary algorithms (EAs) is explained in detail, since this class builds the foundation of the algorithms presented in the subsequent chapters.

### 4.1. Solving Paradigms in Multi-objective Optimization

*Declaration: Parts of this section have been published in [BHSS17]*

#### 4. Methodologies for Solving Multi-objective Optimization Problems

MOO methodology consists of searching for candidate solutions and deciding between different options. Preferences are mostly related to decision-making in this context. The most common approach to characterizing preferences in MOO therefore consists of classifying the order in which searching and deciding take place [CM75, BDMS08].

**A priori technique** methods consider the DM's preferences from the start by taking preference information into account before the search for candidate solutions begins [CCLVV07]. Preference information is used to design or configure tailored algorithms that obtain Pareto optimal solutions that are favored by the DM [BSS11, SBS13, BDDO04, DG11, Das99]. This way, no computational resources are wasted on options that are not interesting to the DM [MA04]. Thereby, given a fixed budget of computation, either more desirable alternatives can be generated or fewer alternatives can be more thoroughly or faster improved. A priori methods, however, usually do not generate a representation of the entire Pareto front. By focusing on specific regions or individual points of the Pareto front, a DM might miss alternatives that could turn out to be interesting if she had greater knowledge of the problem that is solved [BSS15, BHSS17].

**Interactive technique** or progressive optimization methodology acknowledges that preferences may change during the search if new information about the composition of the feasible set of the objective space becomes available [GEF16]. Thereby, equal emphasis is put on searching and deciding. The DM is required to interact with the optimization algorithm by making intermediate decisions to guide it towards solutions she prefers. Interactive technique algorithms allow a greater flexibility in the articulation of preferences, since the DM is allowed to change her opinion during runtime and correct errors if search results do not meet her previous expectations. Constant interaction, however, can also be arduous for the DM and even harmful if she does not fully understand the implications of the choices she makes during the interaction [Mie99].

**A posteriori technique** methods obtain a representation of the entire Pareto front, for example a uniform discretization of a curve for bi-objective problems. The DM is then presented with the entire set of solutions and subsequent MCDA methodology can be applied to pick a candidate for implementation [GEF16]. This way, the search is prioritized to the decision-making. Being able to choose from a set that represents the entire Pareto front bears the advantage that all tradeoffs between different regions of the front can be taken into account when making a decision [Deb01, CCLVV07]. At the same time, however, computing a representation of the entire Pareto front is often costly, since a sufficient number of points must be found to cover it in its entirety. Especially in real-world applications, in which individual function evaluations

#### 4.1. Solving Paradigms in Multi-objective Optimization

may take multiple hours [CSB<sup>+</sup>12, SCBS14], it is often intractable to generate such a representation.

The ontology of Li et al. for characterizing PMOEA [LYBF<sup>+</sup>16, LYBF<sup>+</sup>17] has already been briefly discussed in Section 2.2. It provides multiple concepts to how preferences can be characterized from an algorithmic perspective and incorporates many categorizations that have been proposed throughout the literature [CM75, BKSG15, ASO15]. The ontology groups preference concepts into classes, which are further divided into subclasses. The classes specifically related to preferences in MOO are discussed in the following (see Figure 4.1).

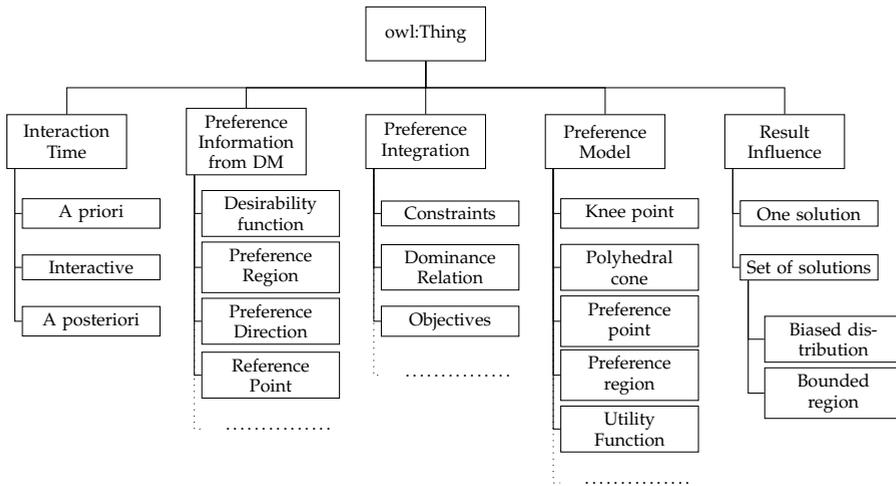


Figure 4.1.: Excerpt of the ontology of Li et al. containing concepts related to preferences – adapted from [LYBF<sup>+</sup>17, Fig. 1]. Listing of classes is limited to concepts relevant to this thesis. Omissions are marked by dotted lines. All classes using the Web Ontology Language (OWL) framework are subclasses of *owl:Thing*<sup>1</sup>. Scalarization functions can be perceived as utility and desirability functions.

A priori technique, interactive technique, and a posteriori technique preference incorporation are summarized under the class *interaction time*. Li et al. adopt and extend the characterization of Bechikh [BKSG15] to define the class of *preference models*. As stated in Section 2.2, this class describes the methodology that is applied to identify preferred solutions. Closely related to this concept is the class *preference information from the DM*, which contains different types of information that a DM must provide for using a corresponding preference model. When using a weighted sum, the DM has to specify weights for the individual objectives, for example. *Preference integration* focuses on how preference models are incorporated into an optimization algorithm [ASO15]. The predicates formulated in Section 2.3 are an example for different integration types. Tradeoff

#### 4. Methodologies for Solving Multi-objective Optimization Problems

optimal solutions can be obtained by either imposing new constraints on the problem, changing the dominance relation or transforming the objective space. Instead of applying these changes at the problem level itself, they can also be handled at the algorithmic level.

The ontology of Li et al. also characterize how preferences influence the search result delivered by the optimization algorithm. The result can either consist of a single solution or an entire set of solutions. Sets of solutions are further divided into biased distributions and bounded regions. Biased distributions focus on obtaining more solutions in preferred parts of the Pareto front, whereas bounded region approaches obtain only points in specific parts of the Pareto front that are preferred by the DM. The algorithmic categorization proposed in this section focuses as well on the *optimization outcome* and is an extension of Li's work.

##### 4.1.1. Closed-form Expression

The Pareto front of a real-valued MOOP of  $m$  objectives is a manifold of at most  $m - 1$  dimensions [Hil01]. A manifold is a geometric structure that locally resembles the Euclidean space, e.g. a curve in one or a surface in two dimensions [Lee10]. Such manifolds can be mathematically described by an implicit function or a parametric equation [Hil01]. Both approaches result in an analytic representation of the Pareto front. Obtaining such a representation of the Pareto front of an MOOP is referred to as *closed-form expression* paradigm. The closed-form expression provides the DM with full knowledge of the Pareto front. She is able to build her preferences on all available information and can choose the alternative she favors most among the entire set of Pareto optimal solutions (see Figure 4.3a).

Finding a closed-form expression of a Pareto front, however, is highly difficult, as there exists no straightforward methodology to derive an exact representation from an arbitrary problem formulation [RW05]. Obtaining an analytic representation of the Pareto front of a real-world MOOP is thereby usually intractable. These problems are often too complex to even analytically determine a single Pareto optimal solution [BSE<sup>+</sup>16, CSB<sup>+</sup>12, SCBS14]. Closed-form expressions are typically only known for artificial problems that are used in the literature to compare and benchmark MOEAs (e.g. [ZDT00, DTLZ05, HHBW06]). It is imperative that the Pareto front of a benchmark problem is known for being able to assess how much effort it takes an algorithm to obtain a suitable representation of the front [HHBW06].

There exist algorithms in MOO that try to approximate a closed-form representation of the Pareto front. Ruzika and Wiecek divide approaches that approxi-

## 4.1. Solving Paradigms in Multi-objective Optimization

mate the Pareto front by (piecewise) polynomial functions into approximation classes. The degree of the polynomial dictates the order of the approximation class – e.g. an approximation by quadratic functions constitutes a second order approximation [RW05]. Ruzika and Wiecek also discuss other function types to approximate a closed-form expression of the Pareto front. Li et al, for example, fit a hyperellipse to the Pareto front to obtain an approximation of a closed-form representation [LFWB03]. In case the Pareto front can be decomposed into segments of specific function types, any method that utilizes these function types for approximation is able to generate an exact representation of the Pareto front [GEF16, Chp. 18]. For example, an algorithm that uses line segments to approximate the Pareto front is able to compute an exact representation of the Pareto front a linear two-objective MOOP. Furthermore, sandwich methods obtain an inner and outer approximation by piecewise functions that serve as upper and lower bound of the Pareto front. The true Pareto front is located between the inner and outer approximation. By tightening the inner and outer approximation an ever more precise representation of the Pareto front can be obtained [RW05, GEF16].

Homotopy-based algorithms are another class of techniques to obtain closed-form expressions of Pareto fronts. These methods are inspired by single-objective parametric optimization (SOPO). In SOPO, the objective function or constraints may possess multiple parameters – function arguments whose values are unknown during optimization and usually only become known during live application. Scalarization functions that are used in MOO to obtain a uniform discretization of the Pareto front (cf. Section 3.1) often possess parameters, for example in the form of weights. By systematically setting and varying these parameters, a uniform discretization of the front may be attained (see the finite set of points paradigm that is discussed next). Homotopy-based algorithms aim at finding a functional description of the Pareto front in the neighborhood of the points obtained by the discretization [Hil01].

### 4.1.2. Finite Set of Points

Since obtaining a closed-form expression of the Pareto front is often intractable, classic MOO methodology aims at approximating the Pareto front by a *finite set of points* [Deb01, CCLVV07]. Such a finite set of points representation usually consists of a uniform discretization of the front. The number of points used for the discretization is specified by the DM. Uniformly distributed points across the Pareto front provide the DM nearly with the same degree of information about the composition of the front as the closed-form expression (see Figure 4.3b). Although the DM is limited to the points obtained by the optimization algorithm in her choice, at least one of these points should satisfy the

#### 4. Methodologies for Solving Multi-objective Optimization Problems

DM's preferences if there are sufficient solutions to choose from and the entire front is covered. The finite set of points notion falls into the category of a posteriori technique approaches.

Finding a uniform approximation of the Pareto front consists of two challenges. First of all, the approximation should be sufficiently close to the actual Pareto front, such that its elements are at least close to Pareto optimal. Ideally, all points of the approximation are located on the actual front. Numeric algorithms that operate on nonlinear functions, however, are often only able to generate solutions that are close to optimality within a certain error bound. This is a general problem in optimization and not specifically related to MOO [Ber99].

Secondly, the approximation should be uniform such that every part of the front is equally covered. The issue is, however, that there does not exist a general definition of uniformity for manifolds in vector spaces [SK97]. Instead, different mathematical concepts and algorithmic techniques are used to achieve uniformity. Scalarization functions such as the weighted sum or the Chebyshev method can be used with uniform sets of weight vectors to compute uniformly distributed points on the Pareto front [DD98, ZL07, LZ09, DJ14]. In general however, uniform weights do not guarantee a uniform discretization of the Pareto front [JD14, DD97]. Heuristic approaches often use niching techniques to force points to equally spread across the Pareto front [DPAM02, ZLT01, ZK04, NDN<sup>+</sup>09]. These techniques penalize points that are close to each other in the objective space, which results in search operators generating more decision vectors that map to areas of the Pareto front that have not been explored, yet. Although niching techniques are capable of spreading points well across the front, they can in general not be used straightaway to decide between two sets of points, which one represents a more uniform approximation. This circumstance prohibits using these techniques to define optimal distributions of points on the Pareto front.

*Hypervolume* [ZT99] is a well-known and one of the few concepts in MOO that allows measuring the uniformity of a Pareto front approximation by a real-valued function. Hypervolume denotes the volume enclosed by a finite set of points with respect to a predefined reference point (see Figure 4.2a). Given a reference point and a fixed number of points, there exists a positioning of these points on the Pareto front such that the hypervolume is maximized, thereby constituting an optimal distribution of points. The contribution of individual points to the overall hypervolume, however, depends on the chosen reference point. This can lead to points changing their optimal position if the reference point is moved, which implies a certain ambiguity in using hypervolume to define an optimal distribution of points [SDS14, IISN17]. Furthermore, computing the hypervolume contribution of an individual point is known to be

### 4.1. Solving Paradigms in Multi-objective Optimization

NP-hard for three and more objectives [BF09]. Finally, a set of points that maximizes hypervolume for a given front may differ from the equidistance notion of a DM (see Figure 4.2b). Hypervolume-based optimization techniques have been successfully applied in MOO [ZK04, BNE07, IHR07, BZ11]

The extended discussion on uniformity serves to illustrate the ambiguity in defining and the difficulty in obtaining good discretized representations of the Pareto front that cover it in its entirety. Thereby, any notion of uniformity for Pareto front approximations in MOO can itself be conceived as a preference. Of course, the finite set of points approach suffers from the same drawback as any a posteriori technique. Obtaining an approximation of the entire Pareto front is computationally expensive. Since this is often neither tractable nor desirable, other paradigms use preferences to restrict the part of the Pareto front that is approximated.

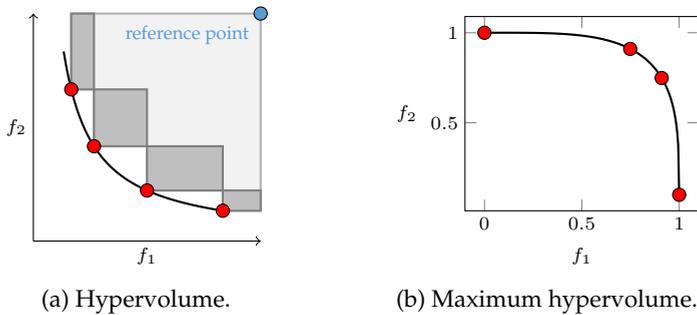


Figure 4.2.: Illustration of the hypervolume concept. The shaded area in Figure 4.2a depicts the volume encompassed by the four red points with respect to the blue reference point. Dark gray areas mark the respective contribution of each individual point to the overall hypervolume. Figure 4.2b shows the Pareto front  $\mathcal{Y}_p$  given by  $\mathcal{Y}_p := \{\mathbf{u} \in \mathbb{R}_{\geq 0}^2 \mid u_1^4 + u_2^4 = 1\}$ . The four red points are positioned on  $\mathcal{Y}_p$  such that they maximize their hypervolume with respect to the reference point (1, 1). The points are not equally distributed across the front, since the two interior points are close to each other and further away from the boundary points.

#### 4.1.3. Subset Approximation

The *subset approximation* paradigm limits the Pareto front approximation to areas on the front that are deemed interesting according to the DM's preferences (see Figure 4.3c) [LYBF<sup>+</sup>16, LYBF<sup>+</sup>17, BKSG15]. All elements in these areas are considered to be of equal desirability to the DM. The distribution of points

## 4. Methodologies for Solving Multi-objective Optimization Problems

within a given area should therefore be uniform. In restricting the approximation to subsets of the Pareto front, the DM is not concerned with alternatives that she would never consider for implementation. The computational resources that are saved this way can be used to improve the subset approximation. At the same time, however, information about the global tradeoffs, which occur between different regions of the Pareto front, is lost. If areas of interest are to be approximated uniformly, the DM faces the same issues as in the finite set of points approach. Identifying preference or knee regions (see Figure 2.2) is an example of subset approximation.

Preferred subsets are often defined by setting a threshold underneath solutions are deemed interesting [SHS10a, BSS11, DG11]. Finding an optimal threshold, however, can be a difficult task. If the threshold is set too high, the subset approximation covers too much of the Pareto front. The DM thereby gains no advantage in focusing on subsets. If the threshold is too low, the subset approximation may consist of too few options or be even empty. Given a fixed threshold, the expanse of the preferred subsets compared to the Pareto front also greatly depends on the curvature and composition of the Pareto front [BSS11]. This makes it difficult for coming up with best practices a priori technique to define meaningful values for the preference thresholds.

Numerous approaches have been developed to compute preferred subsets of the Pareto front. Additional constraints can be enforced on the problem that is solved to curtail the front [SHS10a, BSS11, SHS10b]. Scalarization values can be used to set thresholds that delimit regions of interest [SBS13, DG11]. Weighting methods also use scalarization functions to identify preferred parts of the Pareto front. These approaches insert scalarization values at different steps of finite set of points algorithms. Friedrich, for example, multiplies a weighting coefficient with the niching value in the NSGA-II and SPEA2 algorithms [FKN13]. Brockhoff et al. use weighting coefficients to increase the hypervolume contribution of points that lie in preferred parts of the Pareto front [BBTZ13]. Domination-based approaches enlarge the Pareto cone (see Figure 2.5) leading to areas of the Pareto front becoming dominated. The remaining non-dominated Pareto optimal solutions form the preferred subset [BKS01, SW13, SB13, BSS11].

### 4.1.4. Global Preference Optima

The last paradigm that has been established in MOO is obtaining *global preference optima* (see Figure 4.3d) [Mie99, LYBF<sup>+</sup>16, LYBF<sup>+</sup>17]. Algorithms following this approach compute only a single solution – a solution that is globally preferred by the DM to all other Pareto optimal points. If an MOOP possesses multiple global preference optima, these algorithms usually identify only one of

#### 4.1. Solving Paradigms in Multi-objective Optimization

them. Since these are equally desirable from the DM's perspective, any global optimum represents a best choice from a normative perspective.

Global preference optima eliminate the necessity of choice, since the DM is presented with only one option. All computational resources can therefore be focused on finding a global optimum. This approach, however, does not generate any additional information about the Pareto front. Other, potentially interesting options are not considered [BSS15, BHSS17]. A DM should thereby be well aware of the chosen preference model, its integration into an optimization algorithm and the resulting implications on the search results. For scalarized preferences, these considerations can be made by utilizing the results presented in Chapter 3.

Scalarization is the most common technique to obtain global preference optima. Minimizing a weighted sum (see Definition 33) for a given set of weights, for example, yields a global preference optimum [MA04]. Lexicographic approaches optimize individual objectives in an order specified by the DM. The optimal value of the objective that is most important to the DM is identified first. Subsequently, the optimal value of the second most important objective is determined while requiring that the most important objective maintains its optimal value. This procedure is repeated until all objectives have been optimized [Ehr05]. Reference-point based methods utilize aspiration values for each objective set by the DM. These methods then aim at finding a Pareto optimal point that satisfies the chosen aspiration levels [Mie99]. Knee-point-based algorithms (see Figure 2.2) also belong to the global preference optima paradigm. Such approaches aim at finding points that are located at the greatest convex bulge of the Pareto front and often utilize scalarization techniques [BDDO04, DG11, SBS13, Das99, BSS17].

##### 4.1.5. A Hierarchy of Solving Paradigms

The four paradigms that have been presented build a hierarchy with respect to the knowledge about the Pareto front generated through their application (see Figure 4.4). A closed-form expression allows generating any number of points on the Pareto front. From a finite set of points, subsets can be identified that are favored by a given preference notion. Of all subsets, the globally preferred point can be identified. Note, however, that this hierarchy is not an inclusive relation. Picking the most preferred point of a subset approximation does usually not yield an exact global optimum. At the same time, the subsets identified from an approximation of the entire Pareto front are expected to contain far less elements than the search result of an algorithm that was specifically designed to approximate these subsets. Finally, a closed-form expression cannot be easily discretized into a finite set of uniform points.

#### 4. Methodologies for Solving Multi-objective Optimization Problems

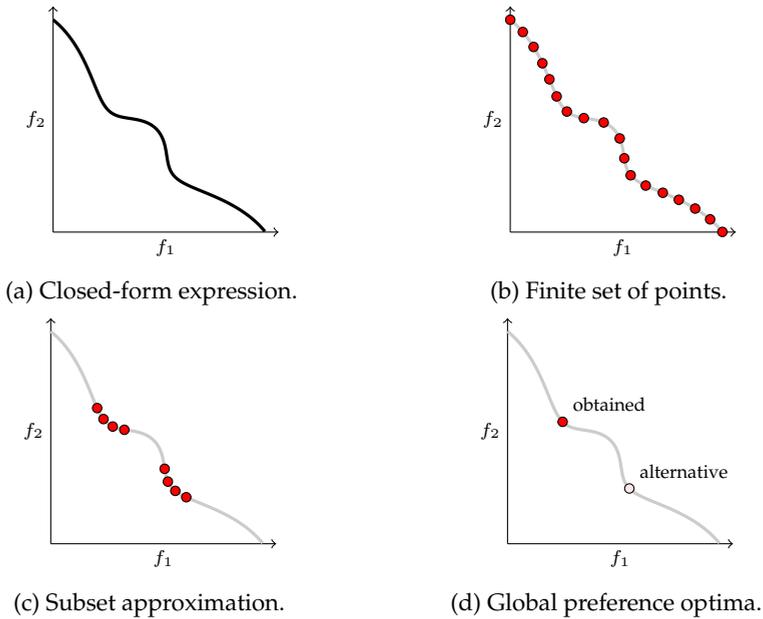


Figure 4.3.: Illustration of solving paradigms in MOO. Figures 4.3a to 4.3d show idealized examples of the result obtained by the respective optimization methodology on the DEB2DK [BDDO04] benchmark problem. Note that algorithms following the global preference optima paradigm usually obtain only one global optimum as search result, marked by *obtained* in Figure 4.3d. Other global optima, marked by *alternative* in Figure 4.3d, are not retrieved. Different runs might obtain different optima.

The hierarchy proposed in Figure 4.4 also orders the paradigms according to the computational effort they require for being executed. Obtaining a closed-form expression of the Pareto front mandates a series of complex computations if curve-fitting or homotopy approaches are applied. These methods usually utilize finite set of points approximations in intermediate steps of their execution [RW05, Hil01]. If a finite set of points approximation is generated by solving multiple instances of a scalarized MOOP, a subset approximation may be obtained by solving only a subset of these scalarized instances. A DM, for example, may choose a small subset of weight vectors for the weighted sum or Chebyshev method [LZ09, JD14]. Subset approximations, however, are often obtained by modifying existing finite set of points algorithms. In this case, a subset approximation may elicit more computational effort if the same number of function evaluations are executed [SHS10a, BSS11, FKN13, BBTZ13]. Global preference optima approaches compute only a single solution and are thereby

## 4.1. Solving Paradigms in Multi-objective Optimization

expected to require less function evaluations [SBS13, Das99, MA04].

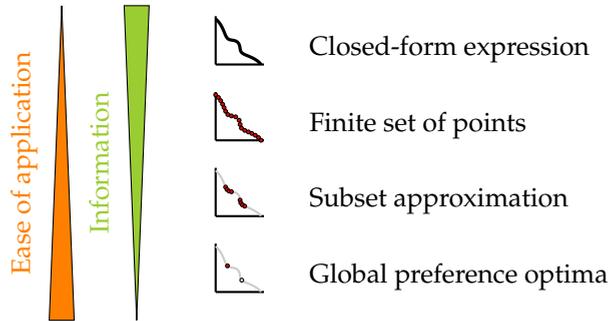


Figure 4.4.: Hierarchy of existing solving paradigms in MOO that describe the outcome an optimization methodology yields. The hierarchical structure expresses how much information about the Pareto front is generated in applying each paradigm. The closed-form expression generates full knowledge of the Pareto front, whereas the global preference optima paradigm attains the least knowledge in only computing a single Pareto optimal point. Conversely, the application of paradigms that generate less information typically requires less computational effort and are thereby easier to apply.

The four paradigms can also be closer regarded in the light of scalarized preference optima. Each approach can be utilized to obtain or approximate the optimum of a scalarized MOOP. The closed-form expression of a Pareto front can be concatenated with the scalarization function that represents the DM's preference. A minimum of the concatenated function is a global optimum of the scalarization function. Obtaining the minimum of the concatenated function requires solving an SOOP, which itself requires the application of a suitable optimization methodology. Scalarization values can be computed for any point of a uniform Pareto front approximation. The point having the smallest scalarization value serves as an approximation of a global scalarization optimum. The same procedure can be applied to the points generated by a subset approximation. Of course, the scalarization optimum must be contained in the preferred region covered by the subset approximation. Additionally, necessary reference points in the objective space for correctly computing scalarization values must be available (see Definitions 36, 39 and 41, for example). Finally, concatenating the objective functions of the MOOP to solve with the scalarization function and obtaining the minimum of the resulting SOOP corresponds to the global preference paradigm.

As it has been suggested in the previous chapters, scalarization values should rather be perceived as guidance in identifying interesting solutions instead

## 4. Methodologies for Solving Multi-objective Optimization Problems

of blindly relying on a global scalarization optimum to be the best choice. Therefore, the suggestion is made to amend the existing hierarchy of solving paradigms by two new approaches that are especially useful in combination with scalarized preferences.

### 4.1.6. Preference-biased Pareto front approximations

*Preference-biased Pareto front approximations* are proposed to strike a balance between the finite set of points and the subset approximation paradigm (see Figure 4.5a). A preference-biased approximation contains more points in regions that are deemed interesting, however still covers the entire Pareto front. This way, the DM attains an insight into the composition of the Pareto front and the global tradeoffs that occur similar to the finite set of points paradigm. At the same time, the DM is presented with more useful alternatives such as in the subset approximation approach. Regions are deemed more interesting if their members are associated with smaller scalarization values [BSS15]. An algorithmic framework for obtaining preference-biased Pareto front approximations using scalarized preference information is presented in the next chapter.

The main idea of the preference-biased Pareto front approximation is that the preference itself should determine the distribution of points across the front. Given a specific preference model, there should exist an optimal allocation of points on the front irrespective of the algorithm chosen for obtaining said allocation. Approaches found in the literature that obtain biased approximations usually do not define a concept of such an optimal allocation. Instead, they rely on niching techniques to bias the distribution of points towards desirable areas [BKS01, BD05, Deb03, BDDO04, SBS13, FKN13]. Many of these approaches also put a higher emphasis on obtaining subsets of the front instead of covering the front in its entirety [BKS01, BD05, BDDO04, FKN13].

Deb [Deb03] proposes modifying the distance metric in the niching mechanism of the NSGA-II algorithm. The modification is referred to as biased sharing approach and allows prioritizing the different objectives. The resulting Pareto front approximations are more dense towards regions that feature small values of highly prioritized objectives. The biased sharing approach, however, does not define an optimal allocation. Shukla et al. [SBS13] have developed an algorithm that splits the number of points dedicated to approximate preferred regions and the rest of the front. If the preferred region is small and the fraction of points dedicated to approximate the preferred region is large, the approximation of the preferred region is more dense compared to the rest of the front. The approach of Shukla et al., however, neither defines an optimal distribution. The biased hypervolume contribution approach by Brockhoff et al. [BBTZ13] defines a density function from which an optimal allocation of points could

## 4.1. Solving Paradigms in Multi-objective Optimization

be derived. Whether the approximation covers all areas of the front, however, depends on the chosen weighting function.

### 4.1.7. Multimodal preference optima

*Multimodal preference optima* is the second paradigm that is proposed within this work. This approach can be considered a compromise between subset approximation and global preference optima. Preferences often allow the identification of multiple regions of interest on a Pareto front. In this context, preferences are characterized as multimodal if those regions of interest are disconnected. It is usually possible to identify a solution in each region that is preferred to all other elements of that specific region. Such points are called local optima and sought in the multimodal preference optima paradigm [BHSS17].

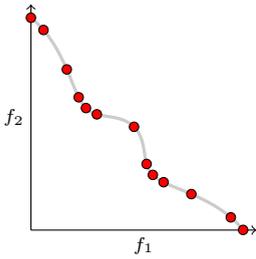
The multimodal preference optima paradigm can be directly applied to scalarization functions. If the domain of a scalarization function is restricted to the Pareto front, the scalarization function may exhibit multiple local and global optima. The multimodal scalarized preference approach consists of computing all optima – local and global – of the scalarization function. Local optima possess the best scalarization value among all Pareto optimal solutions in their neighborhood. They thereby constitute the best choice in their immediate vicinity making them interesting candidates for implementation. An algorithmic approach for computing local scalarization optima is presented in Chapter 6.

Multimodal preference optima approaches are largely untapped in MOO. Existing methodologies that utilize scalarization functions are mainly interested in obtaining a single global preference optimum [Mie99, MA04, DG11, Das99, SBS13]. Branke et al. [BDDO04] have proposed an algorithm that approximates knees on Pareto fronts that feature multiple convex bulges. Their approach generates a tight approximation of the knee region around its strongest bulge, however the corresponding knee point is not explicitly sought. Research on multimodal optimization, however, has received increasing attention in SOO [Pre15]. Methodologies from single-objective multimodal optimization (MMO) can be translated to MOO in the context of scalarization functions (see Chapter 6).

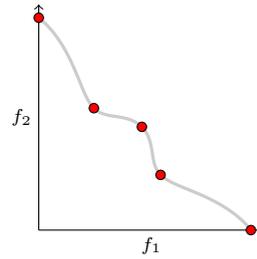
### 4.1.8. An Amended Hierarchy of Solving Paradigms

The complete hierarchy that contains all presented paradigms is depicted in Figure 4.6. Preference-biased approximations are often obtained by introducing new niching mechanism to existing algorithmic frameworks [BDDO04, FKN13,

#### 4. Methodologies for Solving Multi-objective Optimization Problems



(a) Preference-biased approximation.



(b) Multimodal preference optima.

Figure 4.5.: Illustration of new solving paradigms for characterizing optimization outcomes in MOO. Figures 4.5a and 4.5b show idealized examples of the results obtained by the respective optimization methodology on the DEB2DK benchmark problem.

BBTZ13]. Thereby, these approaches elicit a similar computational effort as finite set of points algorithms. Obtaining all global and local preference optima, on the other hand, requires more effort than computing only a single global optimum. Global optimization algorithms are designed to escape and discard local optima if they are encountered during the search [Ber99, Sia16]. A multimodal search strategy explicitly retains local optima and further improves their approximation. If all local optima are to be found, the search space must be sufficiently explored. A global optimum, on the other hand, may be obtained by starting the search from a random point in the search space [Ber99, Sia16].

As stated before, the relationship between the paradigms in Figure 4.4 is not inclusive. The same observation applies to the preference-biased approximation in the amended hierarchy in Figure 4.6. Regions of interest are not uniformly approximated in preference-biased approximations. Points could be eliminated from a finite set of points approximation to obtain a preference-biased approximation. Such a procedure, however, is expected to lead to a very coarse approximation of the corresponding optimal distribution of points on the Pareto front implied by the preference. Furthermore, if the threshold to define regions of interest in the subset approximation paradigm is set too low, approximations to local optima might not be contained in the generated subsets that would otherwise be found using a multimodal optimization procedure [BSS11, BHSS17]. The relationship between multimodal preference optima and global preference optima, however, is strictly inclusive. Any global preference optimum is contained in the set of all local preference optima.

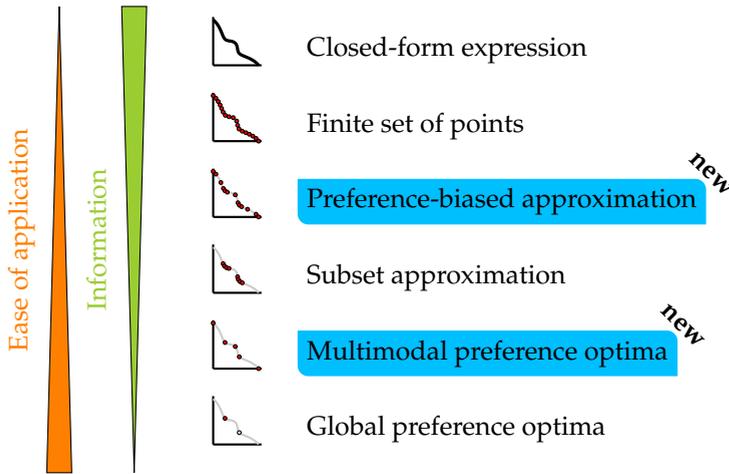


Figure 4.6.: Complete hierarchy of existing and proposed solving paradigms in MOO that describe the outcome an optimization methodology yields.

## 4.2. Nature-inspired Problem Solving

Algorithms for obtaining solutions to MOOPs can be broadly divided into mathematical programming techniques and metaheuristics. Both approaches represent two different schools of thought that are mostly pursued in separate branches of research [GEF16]. Mathematical programming mandates that the MOOP is formulated such that it satisfies a predefined mathematical structure, i.e. it must be given in a canonical form. In nonlinear multi-objective optimization, for example, objective functions and constraints are assumed to be twice continuous and differentiable. If this is the case, it is possible to formulate necessary and sufficient conditions for checking, whether a point is Pareto optimal. These conditions can in turn be exploited to develop algorithms that efficiently search for Pareto optimal solutions. A mathematical programming technique is guaranteed to obtain Pareto optimal solutions within some error bound if the MOOP meets the conditions required by the programming technique that is applied [Mie99, Ehr05].

Metaheuristics are high-level frameworks that define search strategies for applying heuristics to solve optimization problems [Sör15]. Heuristics are solving techniques that rely on stochastic operators to compute solutions. Stochastic operators use (pseudo-)random numbers that are generated during runtime. These random numbers influence the execution of the algorithm and may lead to different results across multiple runs [Deb01, CCLVV07]. In contrast, mathematical programming techniques are deterministic and always yield identi-

#### 4. Methodologies for Solving Multi-objective Optimization Problems

cal results if run with the same configuration. Although metaheuristics have been successfully applied to solve real-world optimization problems, they are never guaranteed to arrive at an optimum and might get stuck at local optima [Sia16, Sör15].

Metaheuristics often mimic natural phenomena or human behavior in obtaining solutions to optimization problems [Sia16, Sör15]. EAs are a broad class of optimization algorithms that simulate the biological process of evolution. They utilize an initial set of points sampled in the domain of the MOOP that is gradually improved using evolutionary operators to converge towards a desired optimum. This set of points that is continually improved is referred to as population. Evolutionary operators generate and improve population members, also called individuals, by mimicking natural selection in the optimization context. A generic framework that illustrates how EAs work as optimization methodology is described in Figure 4.7. The description of EAs in MOO in this section is based on [Deb01, CCLVV07].

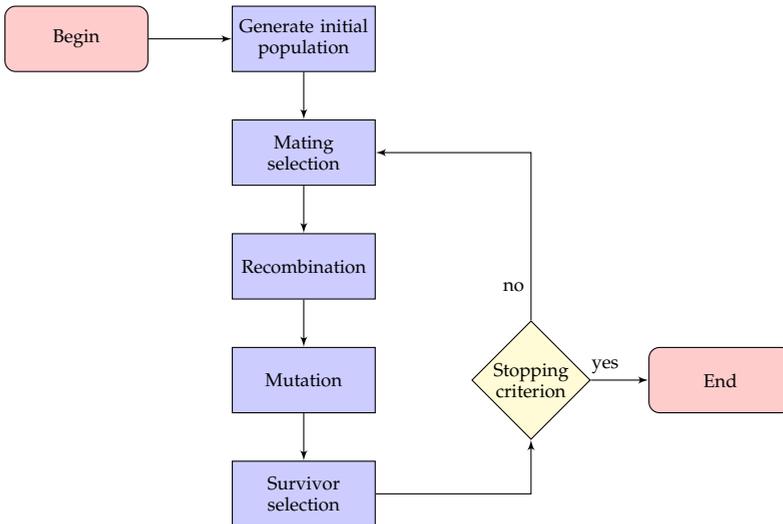


Figure 4.7.: Illustration of the general working principle of an EA. An initial population is generated. In each iteration, individuals are selected from the population into a mating pool and recombined to generate offspring individuals. Offspring individuals are mutated. Only the best population members according to some predefined selection criterion of the offspring generation survive. The surviving members constitute the population of the next iteration. The evolutionary cycle is repeated until a predefined stopping criterion is met.

## 4.2. Nature-inspired Problem Solving

Evolutionary operators can be divided into three categories – selection, recombination (or crossover) and mutation. Selection operators are used to decide which population members procreate to generate offspring individuals. They select and pair population members using stochastic or deterministic choice rules. Recombination operators then use the paired population members – the parents – to generate new offspring individuals by using arithmetic operations to combine the decision vector values of the parents. Recombination is used to explore the search space between the parents. Mutation operators are used to induce minor changes in the decision vectors of offspring individuals. This way, areas of the search space can be explored that are otherwise inaccessible through mere recombination.

In EAs and especially MOEAs, selection operators are also used to decide, which individuals survive to form the population of the next generation. Often, the population of the current iteration and the offspring are combined to a joint set of individuals. The best individuals are picked from the combined set and survive to the next iteration – a concept called elitism. Fitness values are assigned to the members of the combined population that depend on the objective vector of the given individual and the composition of the population. In the finite set of points paradigm, for example, an individual is ranked higher, if it is presumed closer to the Pareto front and if there exist only few neighboring population members in the objective space. Fitness assignment and selection are usually the decisive mechanisms in the finite set of points paradigms by which algorithms differ, since most algorithms rely on the same crossover and mutation operators [DPAM02, ZLT01, ZK04, TFD11].

In evolutionary optimization, population members are further characterized by their geno- and phenotype. The genotype is the representation of the decision vector in the EA. In case of real-valued MOOP, the EA can either operate directly on real-valued variables or use an encoding – reals, for example, can be represented by a binary string to a limited working precision.<sup>2</sup> The phenotype of an individual consists of its objective vector. Recombination and mutation work on the phenotype of an individual, while selection mostly considers the phenotype.

EAs have a long history of successful application in MOO, which is why they are chosen as optimization methodology in this work [Deb01, CCLVV07, GEF16, BDMS08]. Additionally, scalarization functions are regarded as black boxes in this work to give the DM the highest degree of freedom in specifying her preferences. In black box optimization (BBO), no analytical description of the MOOP that is solved is given. No assumptions can thereby be made about the arithmetic structure of the problem that can be exploited, for example to compute

---

<sup>2</sup>It is noted that computers, which are used to solve hard MOOPs, are themselves only able to depict reals to a limiting working precision, since all memory is finite.

#### *4. Methodologies for Solving Multi-objective Optimization Problems*

derivatives [Pre15]. Scalarization functions are often non-smooth or introduce nonlinearities that make the application of mathematical programming difficult [MA04, SBS13, BSS17]. Many of the scalarization functions presented in Section 3.2 also require reference points. These points would be obtained in mathematical programming techniques by formulating and solving separate optimization problems. This, in turn, requires additional expert knowledge from the DM that is normally not directly available. MOEAs, on the other hand, are able to estimate these reference points during runtime [SHS10a, BSS11, SBS13, BSS17, DJ14, JD14].

## 5. Preference-biased Pareto Front Approximations

*Declaration: Parts of this chapter have been published in [BSS15].*

The scalarized preference model assigns a desirability value to every element of the Pareto optimal front. Areas of the front that have larger scalarization values are less desirable compared to areas that feature smaller scalarization values. A Pareto front approximation by a finite set of points should take this preference information into account – regions possessing smaller scalarization values should be more densely approximated, whereas areas of large scalarization values should feature less points. Such a preference-biased approximation provides a DM with enough information to estimate the global tradeoffs that occur on the Pareto front while retaining a tight focus on the areas that are most interesting to him.

This chapter commences by presenting a theoretical foundation for defining an optimal allocation of points given an arbitrary Pareto front and scalarization function. It is shown that such an allocation exists under some very weak requirements. An algorithm that approximates such optimal allocation of points is presented next. The final section of this chapter features a computational study. A quantitative analysis assesses how well the algorithm approximates the optimal preference-biased allocation. A no preference case is also considered in which all Pareto optimal solutions are equally desirable. Therefore, the proposed algorithm is compared to other approaches that compute uniform finite set of points approximations. The results show that the proposed algorithm outperforms these other approaches on many popular benchmark problems. Finally, a qualitative analysis evaluates the usefulness of the preference-biased approximations generated by the algorithm from a decision making perspective.

## 5.1. Theoretical Foundation of Preference-biased Approximations

The scalarized preference model as presented in Chapter 3 assigns numeric values to every Pareto optimal solution. These scalarization values express desirability – smaller values indicate higher and larger values lower desirability. If a Pareto front is discretized into a finite set of points (cf. Sections 4.1.2 and 4.1.6), the discretization should take this preference information into account. More points should be located in areas with smaller scalarization values and less points in regions with higher scalarization values. The scope of this section is presenting a concept that defines an optimal distribution of points across a Pareto front given a specific scalarization function.

The concept for defining optimal distributions of points is founded in the physical phenomenon of electrostatic potential energy. In physics, charged particles exert Coulomb forces onto each other. These Coulomb forces yield electrostatic potential energy. The potential energy that a single charged particle introduces into a physical system with respect to another particle is equal to the product of their charges multiplied by Coulomb’s constant and divided by their Euclidean distance. Given a closed physical system of charged particles, the energy of the system is equal to the sum of all pairwise electrostatic energy between the individual particles. If the number of particles is finite, there exists a positioning of these particles in the system such that the overall energy of the system is minimized [HRW10].

The phenomenon of electrostatic potential energy is translated to the optimization context. The closed physical system is represented by the Pareto front. The charged particles are Pareto optimal solutions. The charge of a particle corresponds to its scalarization value. The distance between particles is equal to the Euclidean distance between points in the objective space. The energy that is produced by a finite set of Pareto optimal points can then be described as in Definition 53.

**Definition 53** (energy [BSS15]). *Let  $S = \{\mathbf{u}^1, \dots, \mathbf{u}^N\} \subseteq \mathcal{Y}_p$  be a finite subset of the Pareto front  $\mathcal{Y}_p$  and  $\Psi$  be a scalarization function. The energy of  $S$  is defined as*

$$U(S) = \sum_{i=1}^{N-1} \sum_{j=i+1}^N \frac{\Psi(\mathbf{u}^i) \cdot \Psi(\mathbf{u}^j)}{\|\mathbf{u}^i - \mathbf{u}^j\|_2}. \quad (5.1)$$

Furthermore,

- $\min_{S \in \mathcal{Y}_p^N} U(S)$  is called energy minimum,
- $\arg \min_{S \in \mathcal{Y}_p^N} U(S)$  is called an  $N$ -optimal distribution for  $\Psi$  on  $\mathcal{Y}_p$ .

## 5.1. Theoretical Foundation of Preference-biased Approximations

The preferred subset of  $X_p$  that is identified by minimizing Equation (5.1) can also be formulated as preference predicate (see Definition 12). The preference-biased approximation predicate (Definition 54) restricts the feasible set to the set of Pareto optimal points that minimize the function  $U$ .

**Definition 54** (preference-biased approximation predicate). *Let  $f = (X, Y, F)$  be the objective function of an MOOP,  $\Psi$  be a scalarization function and  $N > 2$ . The preference-biased approximation predicate is given by  $((X_N, Y, F_N), <_p)$  with*

$$X_N := \left\{ \{\mathbf{x}^1, \dots, \mathbf{x}^N\} \subseteq X_p \mid \arg \min_{\{\mathbf{x}^1, \dots, \mathbf{x}^N\} \in X_p^N} \sum_{i=1}^{N-1} \sum_{j=i+1}^N \frac{\Psi(f(\mathbf{x}^i)) \cdot \Psi(f(\mathbf{x}^j))}{\|f(\mathbf{x}^i) - f(\mathbf{x}^j)\|_2} \right\} \quad (5.2)$$

and

$$F_N := \{(\mathbf{x}, f(\mathbf{x})) \in F \mid \mathbf{x} \in X_N\}. \quad (5.3)$$

There exist noteworthy difference between the physical concept of electrostatic potential energy and its translation to the optimization context. First of all, Coulomb's constant is dropped from Equation (5.1), since it bears no meaning in the optimization context. Multiplying Equation (5.1) by any positive constant does not change the order in which sets are ranked if energy as of Definition 53 is used as a measure for comparison. Secondly, in physics, particles retain their charge if they change their position. In the optimization context, the scalarization value of a point depends on its position in the objective space.

Given a fixed number of points  $N$ , a scalarization function  $\Psi$  and a Pareto front  $\mathcal{Y}_p$ , minimizing Equation (5.1) yields an  $N$ -optimal distribution of points for  $\Psi$  on  $\mathcal{Y}_p$ . Theorem 3 shows that such a distribution exists under very mild conditions. The Pareto front of a real-valued MOOP is compact under very weak restrictions [Hil01]. Even if the Pareto front is disconnected (Figures 5.20e and 5.21b) or discrete, an  $N$ -optimal distribution of points exists. All scalarization functions introduced in Section 3.1 besides the Nash bargaining solution fulfill the requirements of Theorem 3 and Propositions 11 and 12 such that  $N$ -optimal distribution exist for these functions (Corollary 2). The Nash bargaining solution always yields negative scalarization values, which would lead solutions having smaller scalarization values to increase  $U$ , whereas solutions with larger scalarization values would decrease  $U$ . This runs contrary to the assumption that smaller scalarization values imply higher desirability. The Nash bargaining solution, however, can be modified such that it yields only positive values by adding a positive constant that is greater than the optimal scalarization value (Definition 55). The value of the constant influences the position of the individual points at the  $N$ -optimal distribution.

**Theorem 3** (existence of energy minima[BSS15]). *Let  $\Psi$  be a scalarization function,  $\mathcal{Y}_p$  be a Pareto front and  $N > 1$ . Furthermore, let  $\Psi$  be lower semi-continuous on  $\mathcal{Y}_p$ ,*

## 5. Preference-biased Pareto Front Approximations

$\mathcal{Y}_p$  be non-empty and compact and for all  $\mathbf{u} \in \mathcal{Y}_p$  let  $\Psi(\mathbf{u}) > 0$ . Then,  $U$  attains its minimum on  $\mathcal{Y}_p$ .

*Proof.* The domain of  $U$  is  $\mathcal{Y}_p^N$ . The set  $\mathcal{Y}_p^N$  is compact, because  $\mathcal{Y}_p$  is compact. Let  $\varepsilon$  be greater but close to zero and  $Q := \{S \in \mathcal{Y}_p^N \mid U(S) \leq \max_{\mathbf{u} \in \mathcal{Y}_p} N\Psi(\mathbf{u})/\varepsilon\}$  be a lower level set of  $\mathcal{Y}_p^N$ . It follows that  $Q \cap \mathcal{Y}_p^N$  is compact. The function  $U$  is lower semi-continuous on  $Q \cap \mathcal{Y}_p^N$ , since  $\Psi$  is lower semi-continuous on  $\mathcal{Y}_p$  and the denominator of Equation (5.1) never attains the value zero on  $Q$ . Then, the conditions for the Weierstraß extreme value theorem are fulfilled [Ber99].  $\square$

**Proposition 11** (energy minima on disconnected fronts). *Let  $\Psi$  be a scalarization function,  $\mathcal{Y}_p$  be a Pareto front and  $N > 1$ . Furthermore, let  $\mathcal{Y}_p$  be a non-empty union of non-intersecting compact subsets, let  $\Psi$  be lower semi-continuous on each subset and for all  $\mathbf{u} \in \mathcal{Y}_p$  let  $\Psi(\mathbf{u}) > 0$ . Then,  $U$  attains its minimum  $\mathcal{Y}_p$ .*

*Proof.* The proof is analogous to the proof of Theorem 3. If  $\mathcal{Y}_p$  consists of disconnected compact subsets, a lower level set  $Q$  of  $\mathcal{Y}_p^N$  can be constructed such that  $U$  is lower semi-continuous on  $Q \cap \mathcal{Y}_p$ .  $\square$

**Proposition 12** (energy minima on discrete fronts). *Let  $\Psi$  be a scalarization function,  $\mathcal{Y}_p$  be a Pareto front and  $N > 1$ . Furthermore, let  $\mathcal{Y}_p$  be a finite set of points with  $|\mathcal{Y}_p| = K \geq N$  and for all  $\mathbf{u} \in \mathcal{Y}_p$  let  $\Psi(\mathbf{u}) > 0$ . Then,  $U$  attains its minimum  $\mathcal{Y}_p$ .*

*Proof.* Given  $N$  and  $K$ , there exist  $\binom{K}{N}$  different possibilities for choosing a set  $S$  of  $N$  out of  $K$  points. For each such set  $S$  the function  $U$  can be computed.  $\square$

**Definition 55** (modified Nash bargaining solution cf. [BSS15]). *Let  $b \in \mathbb{R}_+$  such that  $b > \max_{\mathbf{u} \in \mathcal{Y}_p} -\Psi^n(\mathbf{u})$ . The modified Nash bargaining solution is defined as*

$$\Psi^{mn}(\mathbf{u}) = \Psi^n(\mathbf{u}) + b. \quad (5.4)$$

**Corollary 2** (cf. [BSS15]).  *$U$  attains its minimum on any Pareto front  $\mathcal{Y}_p$  that is compact, discrete or a union of compact sets if  $\Psi$  is chosen as either the weighted Chebyshev, the modified Nash bargaining solution, angle utility or tradeoff utility; or if  $\Psi$  is chosen as the weighted sum and the weighted product and  $\mathcal{Y}_p \subset \mathbb{R}_{\geq 0}^m$  and  $\mathcal{Y}_p \subset \mathbb{R}_{> 0}^m$  holds, respectively.*

*Proof.* The proof follows from Theorem 3 and Propositions 11 and 12.  $\square$

Examining Equation (5.1), it is possible to assess how the movement of an individual point on the Pareto front affects the amount of energy it introduces into the set  $S$ . Any movement towards an area that has smaller scalarization values is beneficial as it decreases  $U$ . At the same time, however, the denominator of

## 5.2. Algorithmic Approach to Preference-biased Approximations

Equation (5.1) penalizes movements that bring a point closer to other points in the objective space. At the energy minimum, a balance is obtained between locating points in favorable regions of the front and dispersing them across the entire front.

In case the DM possesses no preference with respect to individual solutions and considers all Pareto optimal points equally desirable,  $\Psi$  can be chosen as  $\Psi(\mathbf{u}) := 1$  for all  $\mathbf{u} \in \mathcal{J}_p$ . Such a scenario corresponds to finding a uniform discretization of the Pareto front. Theorem 3 and Propositions 11 and 12 guarantee that such an optimal distribution exists. The results of the computational study in Section 5.3.2 show that choosing  $\mathcal{J}(\mathbf{u}) = 1$  yields distributions that visually appear as uniform.

The notion of finding a preference-biased Pareto front approximation using scalarized preference information can be related to the concept of probability density functions in statistics. The probability density function of a random variable defines the relative likelihood for each value it can take on to occur [FKPT12]. A scalarization function can be interpreted as an inverse density function in this context. A smaller scalarization value would correspond to a larger probability. Drawing multiple values from such a distributions should thereby result in a distribution of points that locates more points in areas of small scalarization values. Brockhoff et al. [BBTZ13] have used such an approach in defining biased hypervolume-maximizing distributions (see Section 4.1.6).

## 5.2. Algorithmic Approach to Preference-biased Approximations

Finding the energy minimum for a fixed  $N$  requires substantial computational resources. If the underlying MOOP, which is solved, possesses  $n$  decision variables, finding the minimum of Equation (5.1) is an optimization problem of  $N \cdot n$  decision variables. Mathematical programming techniques that utilize first and second order derivatives might struggle to find a close approximation of the optimum within a reasonable time frame, as the Jacobian and the Hessian of Equation (5.1) become very large, possessing  $N \cdot n$  and  $(N \cdot n)^2$  entries, respectively. Furthermore as explained in Chapter 3, many scalarization functions require the knowledge of reference points or do not fulfill the Pareto axiom (see Proposition 4). For such scalarization functions, any mathematical programming technique needs to determine such reference points before searching for the minimum of Equation (5.1). Computing the tradeoff utility (see Definition 42) of a single point even consists of solving a separate SOOP. Finding an

## 5. Preference-biased Pareto Front Approximations

$N$ -optimal distribution of points for tradeoff utility thereby mandates solving  $N$  SOOPs for each function evaluations of Equation (5.1). Therefore, an EA is proposed to approximate the minimum of Equation (5.1).

On a side note that further illustrates the complexity of obtaining an  $N$ -optimal distribution of points, finding the minimum of Equation (5.1) is a generalization of the Thomson problem in physics [Tho04]. The Thomson problem is concerned with finding a positioning of a given number of electrons on the unit sphere such that their electrostatic potential energy is minimized. It is equivalent to finding the minimum of Equation (5.1) if  $\Psi := 1$  and  $\mathcal{Y}_p$  is chosen as the unit sphere.<sup>1</sup> Computing solutions to the Thomson problem is known to be difficult. A computer-aided proof for the positioning of five electrons was only published in 2010 [Sch13]. Since Equation (5.1) is a generalization of the Thomson problem for variable charges and arbitrary geometries, finding a minimum of Equation (5.1) is expected to be even harder than solving the Thomson problem for a given  $N$ .

An evolutionary strategy for approximating the energy minimum faces two challenges. First of all, the population of the EA needs to converge towards the energy minimum state. Secondly, once a good approximation has been attained, the algorithm should retain said approximation and avoid random fluctuations in the positioning of solutions across the Pareto front in subsequent generations. The first challenge can be further divided into two subtasks: the population needs to converge towards the front and subsequently disperse on the front to approximate the energy minimum. An archive-based steady state algorithm is proposed to tackle these two challenges. Archive-based EAs maintain a set of candidate solutions that is separate from the current population. An archive usually contains the best candidate solutions encountered so far during the search [CCLVV07]. Steady state algorithms produce only one offspring in each iteration [CCLVV07]. The archive contains those elements that form the currently best known approximation of the energy minimum. New candidate solutions are only accepted into the archive if they improve the current approximation. The approximation is either improved by candidate solutions moving closer to the front or further away from each other. Thereby, the archive is expected to gradually converge towards the front and the energy minimum state, meeting the first challenge. The steady-state approach implies that only one candidate solution at a time is tested for eligibility to enter the archive. The archive will therefore never return to a previous composition of candidate solutions avoiding random fluctuations.

An outline of the ESPEA is presented in Algorithm 1. Note that population

---

<sup>1</sup>The unit sphere, however, is not a Pareto front, since there exist sphere elements that Pareto dominate other sphere elements. The sphere segment located in the negative orthant, for example, dominates the unit sphere segment in the positive orthant.

## 5.2. Algorithmic Approach to Preference-biased Approximations

members are elements of the search space. To facilitate a simplified notation the objective vector of any population member  $\mathbf{x}^i$  is referred to as  $\mathbf{f}^i := f(\mathbf{x}^i)$ .<sup>2</sup> The algorithm starts by generating an initial set of candidate solutions (Line 1) from which all elements, whose images are not Pareto-dominated are copied into the archive  $A$  (Line 2). Next, the algorithm repeats a loop until a predefined stopping criterion is met. At the beginning of each iteration, a single new candidate solution  $\mathbf{x}$  is generated using evolutionary operators (Line 4)<sup>3</sup>. All archive members, whose images are Pareto-dominated by  $f(\mathbf{x})$  are eliminated from the archive (Line 5). In case  $f(\mathbf{x})$  is not Pareto-dominated by any archive members and does not possess the same objective values as any existing archive member, it is checked whether  $\mathbf{x}$  is eligible to join the archive. In case the archive has not reached its maximum desired size of  $N$ ,  $\mathbf{x}$  is added to the archive (Line 8). Otherwise, the algorithm checks, whether  $\mathbf{x}$  replacing any other archive member leads to a reduction of the total energy of the archive.

The function  $e(\mathbf{y})$  in Line 10 computes the amount of energy individual  $\mathbf{y}$  introduces into the archive  $A$ :

$$e(\mathbf{y}) = \sum_{\mathbf{z} \in A \setminus \{\mathbf{y}\}} \frac{\Psi(f(\mathbf{y})) \cdot \Psi(f(\mathbf{z}))}{\|f(\mathbf{y}) - f(\mathbf{z})\|_2}. \quad (5.5)$$

For any  $i \in [N]$ , entry  $e_{-\mathbf{y}^i}(\mathbf{x})$  in vector  $\mathbf{e}$  (Line 11) states the energy  $\mathbf{x}$  would introduce into  $A$  if  $\mathbf{x}$  was to supersede  $\mathbf{y}^i$ :

$$e_{-\mathbf{y}^i}(\mathbf{x}) = \sum_{\mathbf{z} \in A \setminus \{\mathbf{y}\}} \frac{\Psi(f(\mathbf{x})) \cdot \Psi(f(\mathbf{z}))}{\|f(\mathbf{x}) - f(\mathbf{z})\|_2}. \quad (5.6)$$

The replacement of an existing archive member takes place in Line 12. It might be the case that there exist multiple archive members, whose replacement by  $\mathbf{x}$  would yield a reduction in the overall energy of the archive. In this scenario, different replacement strategies could be devised. The following three strategies are proposed. Let  $A_{>}$  be the subset of  $A$  that consists of those members that introduce more energy into the archive than  $\mathbf{x}$ , i.e.  $A_{>} := \{\mathbf{y} \in A \mid e_{-\mathbf{y}}(\mathbf{x}) < e(\mathbf{y})\}$ .

**Best feasible position (BFP)**  $\arg \min_{\mathbf{y} \in A_{>}} e_{-\mathbf{y}}(\mathbf{x})$

The offspring  $\mathbf{x}$  is inserted into the archive such that it introduces the least energy among all elements it can replace in  $A_{>}$ .

<sup>2</sup>To avoid confusion, the short notation  $\mathbf{f}^i := f(\mathbf{x}^i)$  is only used if there exist no other indexed candidate solutions in the current context, e.g.  $\mathbf{y}^i$ , such that the mapping of  $\mathbf{f}^i$  is distinct.

<sup>3</sup>The procedure of how new individuals are generated using evolutionary operators is described in Section 5.3.1.1

## 5. Preference-biased Pareto Front Approximations

**Worst in archive (WIN)**  $\arg \max_{\mathbf{y} \in A_{>}} e(\mathbf{y})$

Individual  $\mathbf{a}$  supersedes the archive member that exhibits the largest energy of all elements in  $A_{>}$ .

**Largest energy decrease (LED)**  $\arg \max_{\mathbf{y} \in A_{>}} (e(\mathbf{y}) - e_{-\mathbf{y}}(\mathbf{x}))$

The difference in archive energy before and after replacement is maximized.

The three replacement mechanisms pursue different conceptual ideas. WIN focuses on eliminating the worst archive members in terms of energy contribution, while BFP inserts  $\mathbf{x}$  in the archive such that its effect on the overall archive energy is minimal. Both strategies, however, do not consider the actual energy decrease caused by  $\mathbf{x}$  replacing an existing archive member as in the case of LED. The absolute energy decrease that occurs by applying BFP and WIN might only be marginal. On the other hand, focusing on the energy decrease might not lead to finding suitable replacements for members that introduce the highest energy into the archive. All three strategies are compared in a computational study in Section 5.3.1.

---

**Algorithm 1:** Electrostatic Potential Energy Evolutionary Algorithm [BSS15]

---

**Input :** MOOP  $f$ , replacement strategy *strategy*

**Output:** Pareto front approximation  $A$

```

1 Generate and evaluate initial population  $P$ 
2  $A := \{\mathbf{y} \in P \mid \nexists \mathbf{z} \in P : f(\mathbf{z}) \prec_p f(\mathbf{y})\}$ 
3 repeat
4   Generate and evaluate a single new candidate solution  $\mathbf{x}$ 
5    $A := \{\mathbf{y} \in A \mid f(\mathbf{x}) \not\prec_p f(\mathbf{y})\}$ 
6   if  $\forall \mathbf{y} \in A : (f(\mathbf{y}) \not\prec_p f(\mathbf{x})) \wedge (f(\mathbf{x}) \neq f(\mathbf{y}))$  then
7     if  $|A| < N$  then
8        $A := A \cup \{\mathbf{x}\}$ 
9     else
10      Calculate  $e(\mathbf{y})$  for all  $\mathbf{y} \in A$ 
11      Calculate  $\mathbf{e} := (e_{-\mathbf{y}^1}(\mathbf{x}), \dots, e_{-\mathbf{y}^N}(\mathbf{x}))$ 
12       $\text{replace}(A, \mathbf{x}, \text{strategy})$ 
13 until stopping criterion
14 return  $A$ 

```

---

ESPEA's approach to approximating an  $N$ -optimal distribution of points works twofold. The archive of non-dominated individuals ensures that the population gradually converges towards the Pareto front. By retaining only non-dominated individuals, a strong selection pressure is put on quickly converging towards the front. Maintaining an archive of only non-dominated individuals

## 5.2. Algorithmic Approach to Preference-biased Approximations

has been successfully applied in other EAs [TFD11]. The replacement strategies are niching mechanism for moving the population towards the  $N$ -optimal distribution on the front once the desired archive size has been reached. By generating only a single new individual instead of multiple individuals in each iteration, the archive is expected to converge quicker to an energy minimum state. In general, steady state algorithms are known to require less function evaluations for obtaining a representation of the Pareto front compared to generational algorithms, whose offspring generations usually have the same size as the current population [DNLA09]. The steady state approach has also been successfully employed to obtain approximations of a hypervolume (see Figure 4.2) maximizing distribution of points. SMS-EMOA, for example, utilizes topological sorting (see Figure 2.4) and eliminates the individual in the last front that has the smallest hypervolume contribution [BNE07].

The computational complexity of ESPEA is analyzed next. The Bachmann-Landau [Cor09] notation is used for describing the worst case runtime of the algorithm. The analysis describes the runtime of a single iteration of the algorithm and counts basic arithmetic operations as computation steps. Removing dominated members from the archive elicits an effort of  $\Theta(Nm)$ . In case the archive is full, all of its  $N$  members need to be checked for elimination. A single check for Pareto dominance requires at most  $m$  arithmetic operations, since  $m$  objective values are possibly compared. During the removal of dominated archive members, it can be checked, whether  $f(\mathbf{x})$  itself is dominated or possesses the same objective values as another archive member. No additional effort is required. If the archive has not reached its full size,  $\mathbf{x}$  can be added in constant time. Equation (5.5) in Line 10 needs to be computed for each archive member. Scalarization values can be precomputed once for every archive member and  $\mathbf{x}$  resulting in a runtime of  $\Theta(N \cdot O(\Psi))$ . Computing the Euclidean distance between two objective vectors requires an effort of  $\Theta(m)$ . Computing Equation (5.5) for a single archive member thereby elicits an effort of  $\Theta(Nm)$  and for the entire archive  $\Theta(N^2m)$ .

When computing Equation (5.5) sequentially for all archive members  $\{\mathbf{y}^1, \dots, \mathbf{y}^N\}$  intermediate results can be stored to save additional effort. The addend  $\Psi(\mathbf{f}^i) \cdot \Psi(\mathbf{f}^j) / \|\mathbf{f}^i - \mathbf{f}^j\|$ , for example, is both contained in  $e(\mathbf{y}^i)$  and  $e(\mathbf{y}^j)$ , since the energy between any two individuals is pairwise symmetric. Consequently, only  $N(N - 1)/2$  fractions in Equation (5.5) need to be computed instead of  $(N - 1)^2$ . Still, computing  $N(N - 1)/2$  fractions results in the same complexity class of  $\Theta(N^2m)$ . Storing the fractions increases the space complexity of ESPEA to  $O(N^2)$ . Additionally, summing up all fractions for a single archive member requires  $N - 1$  steps implying a total effort of  $\Theta(N^2)$  for all archive members.

The computation of vector  $\mathbf{e}$  in Line 11 requires an effort of  $\Theta(Nm)$ . As for Equation (5.5), the fractions in Equation (5.6) can be precomputed once for each

## 5. Preference-biased Pareto Front Approximations

pair of  $\mathbf{x}$  and all other archive members, eliciting an effort of  $\Theta(Nm)$ . For any  $\mathbf{y}^1, \mathbf{y}^2 \in A$  it holds that

$$e_{-\mathbf{y}^1} = e_{-\mathbf{y}^2} - \frac{\Psi(f(\mathbf{x})) \cdot \Psi(\mathbf{f}^i)}{\|f(\mathbf{x}) - \mathbf{f}^i\|_2} + \frac{\Psi(f(\mathbf{x})) \cdot \Psi(\mathbf{f}^j)}{\|f(\mathbf{x}) - \mathbf{f}^j\|_2}. \quad (5.7)$$

It follows that computing  $e_{-\mathbf{y}^1}$  necessitates an effort of  $\Theta(N)$ , while all subsequent entries  $e_{-\mathbf{y}^2}, \dots, e_{-\mathbf{y}^N}$  can be computed in constant time. Applying any of the presented replacement strategies in Line 12 requires  $\Theta(N)$ , since finding the minimum of a set of  $N$  elements requires  $N - 1$  comparisons. If the WIN strategy is applied, Line 11 may be skipped. The total effort for executing a single ESPEA iteration then results in a runtime of  $\Theta(N^2m)$ .

The runtime of ESPEA can be further reduced if results of previous iterations are kept. If  $\mathbf{x}$  is not added to the archive, the values computed in Line 10 can be retained. Even if  $\mathbf{x}$  replaces an existing archive member the following update scheme can be used to reduce the overall runtime to  $\Theta(Nm)$ . Let  $e^t(\mathbf{y})$  denote the function value  $e(\mathbf{y})$  of archive member  $\mathbf{y}$  at iteration  $t$  and let  $\mathbf{z}$  be the archive member that is replaced by  $\mathbf{p}$ . Then,

$$e^{t+1}(\mathbf{y}) = e^t(\mathbf{y}) + \frac{\Psi(f(\mathbf{x})) \cdot \Psi(f(\mathbf{y}))}{\|f(\mathbf{x}) - f(\mathbf{y})\|_2} - \frac{\Psi(f(\mathbf{z})) \cdot \Psi(f(\mathbf{y}))}{\|f(\mathbf{z}) - f(\mathbf{y})\|_2}. \quad (5.8)$$

The update mechanism described in Equation (5.8) elicits a constant effort for each archive member and thereby a total effort of  $\Theta(N)$  for the entire archive. Additionally, it holds that  $e^{t+1}(\mathbf{x}) = e_{-\mathbf{z}}(\mathbf{x})$ . The update mechanism in Equation (5.8) can also be utilized, whenever  $\mathbf{x}$  eliminates an archive member by Pareto-domination in Line 5. In case the entire archive is dominated by  $f(\mathbf{x})$ , this would result in an effort of  $\Theta(N^2)$ , since  $N$  updates are performed consisting of  $N - 1, \dots, 1$  steps as the archive becomes smaller. On average, however, ESPEA's runtime can be reduced to  $\Theta(Nm)$ , since in later stages of the search, when the archive is close to the Pareto front, the image of a newly created individual is only expected to dominate a few existing archive members. Thereby, ESPEA is faster than other popular steady-state MOEAs such as NSGA-II [DNLA09], SPEA2 [DNLA09] or SMS-EMOA [BNE07], whose worst-case runtime complexity is  $\Theta(N^2m)$ ,  $\Theta(N^2 \log N)$  and  $O(2^m)$  (for  $m > 3$ ), respectively. Table 5.1 contains an overview of the time complexity of the different computation steps of ESPEA.

ESPEA can further be modified to constrain the preference-biased approximation to a subset of the Pareto front by changing the dominance relation in Lines 5 and 6. Any dominance relation such as polyhedral cones (Definition 22), variable cones [SB13] or tradeoff domination (Definition 16 and others can be utilized. Examples are provided in Section 5.3.3.

## 5.2. Algorithmic Approach to Preference-biased Approximations

Table 5.1.: Overview of the time complexity of ESPEA of a single iteration. Let  $k$  denote the number of archive members that are eliminated in Line 5. The column  $k \ll N$  considers the case if the number of eliminated archive members is negligible in comparison to the population size and describes the expected behavior of ESPEA when the archive is close to the Pareto front. In the naive column, ESPEA does not retain any information about the function values of  $e$  and  $e_{-y}$ , whereas in the retain column, ESPEA uses the update strategy of Equation (5.8).

Step	Naive		Retain	
	Worst	$k \ll N$	Worst	$k \ll N$
Remove dominated individuals from archive (Line 5)	$\Theta(Nm)$	$\Theta(Nm)$	$\Theta(N^2)$	$\Theta(Nm)$
Calculate $e(\mathbf{a})$ for all archive members (Line 10)	$\Theta(N^2m)$	$\Theta(N^2m)$	$\Theta(N)$	$\Theta(N)$
Calculate $e$ (Line 11)	$\Theta(N)$	$\Theta(N)$	$\Theta(N)$	$\Theta(N)$
Archive update (Line 12)	$\Theta(N)$	$\Theta(N)$	$\Theta(N)$	$\Theta(N)$
Complete iteration	$\Theta(N^2m)$	$\Theta(N^2m)$	$\Theta(N^2)$	$\Theta(Nm)$

On a final note, it needs to be mentioned that electromagnetism-inspired optimization algorithms in MOO have been the subject of research in the past [TK06, CAA14, CAA15, HJL15]. These approaches, however, have translated the physical phenomenon of Coulomb forces to search operators instead of defining an optimal distribution of points on the Pareto front, which is sought by ESPEA. Said algorithms thereby follow the finite set of points paradigm (Section 4.1.2) instead of considering preference information to obtain biased Pareto front approximations.

Tsou and Kao [TK06] are credited as having developed the first electromagnetism-inspired heuristic for MOOPs—MOEM—which is itself an extension to a single-objective optimization algorithm—EM heuristic [BF03]. The EM heuristic is a population-based approach that assigns to each candidate solution a charge, which depends on its deviation from the current estimate of the globally optimal objective value. The charges are used to compute forces that move the particles through to the decision space. Particles that possess smaller objective values attract particles having larger objective values, while candidate solutions exhibiting larger objective values repel particles with smaller objective values. EM further employs a local search strategy in each iteration that probes the particles' neighborhood to speed up the convergence.

The EM heuristic is translated by Tsou and Kao to MOO by introducing the

## 5. Preference-biased Pareto Front Approximations

following modification: Instead of considering the deviation from the current estimate of the globally optimal value, the particle's distance to the closest population member that is not dominated by any other candidate solution is used to compute the charge value. MOEM maintains an archive of nondominated solutions that are found during the search. When the archive has exceeded its maximum size, random archive members are deleted until the archive retains its maximal size.

Carrasqueira et al. [CAA14] have further improved the MOEM algorithm by Tsou and Kao. They utilize the crowding distance metric (Algorithm 3) to truncate the archive and introduce changes to the charge computation and particles' movement resulting from the force vectors to speed up the convergence. Subsequent modifications of MOEM by Carrasqueira et al. [CAA15] have focused on improving the local search strategy. There exist further multi-objective extensions of the original EM algorithm. Han et al. use the number of candidate solutions a particle dominates as its charge value [HJL15]. Other approaches to EM-inspired optimization algorithms have focused on specific types of optimization problems such as flow shop scheduling [Kha14] or vehicle routing [YE10].

### 5.3. Computational Analysis of Preference-biased Approximations

The computational analysis is split into three parts. The first part consists of a comparison of the three replacement strategies BFP, WIN and LED. The second part of the analysis is concerned with the no preference case, in which all Pareto optimal solutions are considered to be equally desirable. ESPEA is compared to other MOEAs that aim at obtaining a uniform finite set of points approximation of the Pareto front. The third and final part features a qualitative analysis of the preference-biased Pareto front approximations generated by using ESPEA with scalarized preference information.

All studies were conducted within the jMetal framework version 4.5 [DN11]. jMetal is a Java-based software for developing, testing and applying algorithms to solve MOOs. The framework provides implementations of existing optimization algorithms, optimization problems and search operators. There exist multiple abstraction layers to represent concepts from optimization theory such as evolutionary operators, populations or candidate solutions, which allow easy development of custom code. ESPEA was developed within the jMetal framework and the code is publicly available online [Bra].

### 5.3.1. Quantitative Comparison of Replacement Strategies

The comparison of the replacement strategies BFP, WIN and LED features two different simulation studies. The first experiment examines how the energy (Equation (5.1)) of ESPEA's archive evolves from a set of randomly drawn decision vectors if the search space is restricted to the set of Pareto optimal solutions  $X_p$ . The second experiment assesses how close each approximation produced by the three different replacement strategies is to the optimal distribution of points if the original feasible set is considered as search space.

Both studies require the knowledge of reference  $N$ -optimal distributions of points that serve as benchmark for allowing a quantitative comparison of the three replacement strategies. Such reference  $N$ -optimal distributions can only be computed with sufficient accuracy if  $N$  is not too large. A size of 50 was found to produce sufficiently accurate reference distributions, whereas larger values would not. For this reason, a population size of 50 was chosen instead of 100, which is a common value found in the literature. Each run of the different replacement strategies was initialized with the same random population. Decision variables were drawn from a uniform distribution in the first experiment and latin hypercube sampling [MBC79] in the second experiment. Uniform sampling was chosen in the first study, since latin hypercube sampling produces close to optimal distributions if the search space is restricted to the Pareto optimal set. Latin hypercube sampling generally allows a better exploration of the search space in early iterations. In the following, search operators, test problems and performance indicators are introduced before the simulation results are presented and discussed. A summary of the study setup is provided in the appendix in Table E.1.

#### 5.3.1.1. Replacement Strategies Study Search Operators

Two different crossover operators were used within ESPEA depending on the current archive size. When the archive had not reached its maximum size, Simulated Binary Crossover (SBX) [AD94] was used. SBX is a recombination technique that simulates the crossover of binary strings on real-valued decision variables. In MOO, a binary string is a decision variable that is represented by a sequence of zeros and ones. Given two binary strings, single-point binary crossover randomly chooses a cutoff point and exchanges the substrings of the two strings to generate offspring individuals (see Figure 5.1). If binary strings are used to encode real-valued numbers, offspring individuals can be characterized by the distance to their parents in the search space – the spread factor. Considering all feasible sequences of zeros and ones of a fixed length and all possible cutoff points, a probability density function for the spread factor can

## 5. Preference-biased Pareto Front Approximations

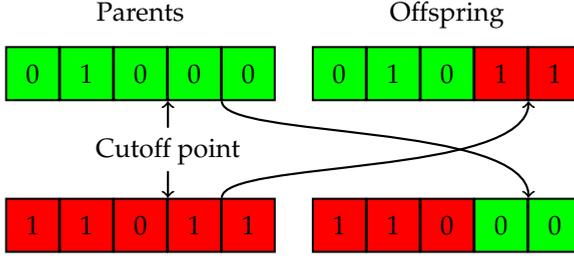


Figure 5.1.: Illustration of single-point binary crossover. Binary strings are divided at the cutoff point. Substrings are swapped between parents to generate offspring.

be computed. Said density function describes the distance of offspring to their parents depending on the chosen cutoff point. SBX recombines real numbers such that the resulting offspring individuals fit the density function for binary crossover. For SBX, there further exists a distribution index  $\eta_c$  that controls the shape of the density function. Small distribution indices lead to offspring being located further away from their parents, whereas large indices result in offspring being close to their parents [AD94]. A crossover probability of 1.0 and a distribution index of 20 were chosen, which have empirically shown to generate good search results in MOO [DPAM02, ZK04, BNE07]. While crossover probabilities of 0.9 are more common in the literature, preliminary tests did not reveal a significant performance difference the values 0.9 and 1.0. Parents for crossover were chosen randomly from the archive.

The mathematical description of SBX is summarized as follows. Let  $\mathbf{x}^o$  denote the offspring of individuals  $\mathbf{y}^1$  and  $\mathbf{y}^2$  and let  $\mathbf{r} \in [0, 1]^n$  be a vector of uniform random numbers. Every component  $x_i^o$  of  $\mathbf{x}^o$  is computed by the following formula:

$$x_i^o = 0.5 \left( (1 + \beta_i) y_i^1 + (1 - \beta_i) y_i^2 \right), \quad (5.9)$$

where

$$\beta_i = \begin{cases} (2r_i)^{\frac{1}{\eta_c+1}} & \text{if } r_i \leq 0.5 \\ \left( \frac{1}{2(1-r_i)} \right)^{\frac{1}{\eta_c+1}} & \text{else.} \end{cases} \quad (5.10)$$

Whenever ESPEA's archive possessed its maximum size, differential evolution (DE) [SP97] was applied as crossover operator. DE comprises a family of crossover operators that uses three or more parents for recombination. These

### 5.3. Computational Analysis of Preference-biased Approximations

operators generate a mutant decision vector by linearly combining parent decision vectors. Recombination takes place by randomly choosing between entries from the mutant vector and another parent decision vector to form a new offspring decision vector. DE operators mainly differ from each other by how the mutant vector is computed and how the entries from the mutant and parent vector are chosen. The *current-to-rand/1/bin* recombination scheme was applied, which has been empirically verified to exhibit good performances [MMVRCC06]. Let  $\mathbf{x}$ ,  $\mathbf{y}^1$ ,  $\mathbf{y}^2$ ,  $\mathbf{y}^3$  denote parent decision vectors and  $c_k, c_f \in [0, 1]$ . The mutant vector  $\mathbf{x}^m$  is computed in the following way:

$$\mathbf{x}^m = \mathbf{x} + c_k(\mathbf{y}^3 - \mathbf{x}) + c_f(\mathbf{y}^1 - \mathbf{y}^2). \quad (5.11)$$

The parameters  $c_k$  and  $c_f$  are combination coefficients that control the weight of the different parents on the mutant vector and 0.5 was chosen as their value, which is commonly used in the literature. Let  $p_c$  denote the recombination probability,  $\mathbf{r} \in [0, 1]^m$  denote a vector of uniform random numbers and  $j \in [m]$ . The offspring vector  $\mathbf{x}^o$  is computed as follows:

$$x_i^o := \begin{cases} x_i^m & \text{if } r_i < p_c \text{ or } i = j \\ x_i & \text{else.} \end{cases} \quad (5.12)$$

The index  $j$  guarantees that the offspring vector  $\mathbf{z}$  contains at least one entry of the mutant vector  $\mathbf{x}^m$  irrespective of the chosen values for  $c_k$  and  $c_f$ . A recombination probability of  $p_c = 0.5$  was chosen.

SBX has been successfully applied in many MOEAs to obtain finite set of points approximations of the Pareto front [DPAM02, ZLT01, ZK04, TFD11], which is why it was chosen in this study. Differential evolution has been shown to outperform SBX as search operator [TF07, LZ09]. As long as the archive is still small, differential evolution requiring four instead of two parents for crossover leads to the same parents being more frequently recombined in early stages of the search. This, on the other hand, might reduce the genetic diversity of the archive and lead to an insufficient exploration of the search space. Therefore, differential evolution is only applied once the archive has reached its full size. Parents are selected randomly, such that equal weight is given to improving the entire distribution of points. The archive replacement strategies guarantee that the scalarized preference information is taken into account when new elements enter the archive.

Polynomial mutation [DG96] was applied to all offspring irrespective of the current archive size. The perturbations introduced by polynomial mutation to the offspring decision vector are drawn from a polynomial density function. Similar to the working principle of SBX, polynomial mutation also utilizes a distribution index  $\eta_m$  to control the spread factor of the mutated offspring. For any  $i \in [m]$  let  $x_i^l$  and  $x_i^u$  denote the lower and upper bound of  $x_i$ ,  $\mathbf{r}, \mathbf{s} \in [0, 1]^m$

## 5. Preference-biased Pareto Front Approximations

be two vectors of uniform random numbers and  $p_m \in [0, 1]$  be the mutation probability. The mutated decision vector  $\mathbf{x}^p$  of any  $\mathbf{x} \in \mathbb{R}^m$  is computed in the following way.

$$x_i^p = \begin{cases} x_i + \rho(x_i^u - x_i^l) & \text{if } r_i < p_m \\ x_i & \text{else,} \end{cases} \quad (5.13)$$

with

$$\rho = \begin{cases} (2s_i)^{\frac{1}{\eta_m+1}} & \text{if } s_i < 0.5 \\ 1 - (2 - 2s_i)^{\frac{1}{\eta_m+1}} & \text{if } s_i \geq 0.5. \end{cases} \quad (5.14)$$

The mutation probability was set to  $p_m = 1/m$  and the distribution index to  $\eta_m = 20$ , which are values that are both recommended in the literature [DPAM02, ZLT01, ZK04].

### 5.3.1.2. Replacement Strategies Study Test Problems

The literature on MOO has produced numerous benchmark problems for testing and comparing MOOAs. These benchmark problems aim at recreating challenges that are possibly encountered in real-world optimization problems, however their objective functions are usually not related to any real-world quantities. Test problems must be carefully selected such that they are representative for the target application for which the algorithms are tested.

Assessing the performance of optimization algorithms by comparing their performance on benchmark problems must be discussed in light of the no free lunch theorem for optimization [WM97]. The no free lunch theorem implies that the average performance of any two algorithms is the same if all conceivable optimization problems are considered. This means that the computational results of this and all subsequent chapters are not generalizable to all possible optimization problems. An algorithm that is shown to perform well on the problems considered in this work is not expected to outperform other algorithms on a randomly picked MOOP. Instead, the problems considered in the experiments of this work are expected to be representative of the type of problems to which the algorithms are applied in practice.

MOO refers to problems possessing two or three objective functions, whereas problems with more objectives are referred to as many-objective optimization problem (MaOP) [ITN08]. ESPEA is mainly intended to solve MOOPs. Preference-biased Pareto front approximations are most beneficial to a DM if she can easily comprehend the impact of her preferences on the distribution of points. Comprehension is furthered by visualization and visualizing Pareto fronts of four or more objectives is usually very difficult.

### 5.3. Computational Analysis of Preference-biased Approximations

An important criterion in selecting benchmark problems for this study was that an analytical description of the Pareto front must be known. The different replacement strategies are compared by assessing how well they approximate the energy minimum of Equation (5.1). Therefore, for each problem a distribution of points that is sufficiently close to the energy minimum, which serves as benchmark, must be obtained. Such distributions can be generated using mathematical programming techniques if there exists an analytical representation of the front [BSS15]. Secondly, problems were selected based on the curvature and composition of the Pareto front. The geometry of the Pareto front has a large effect on the optimal positioning of points at the energy optimum. Therefore, two and three objective problems with convex, concave and mixed curvatures were chosen for the study.

The problem ZDT1 has a strictly convex and ZDT2 a strictly concave Pareto front [ZDT00] (Figure 5.2). ZDT1's front is tangential to the  $f_2$ -axis towards the left extreme point and ZDT2 is tangential to the vertical line passing through the point  $(0, 1)$  towards the left extreme point. The ZDT problem family has been frequently used in the past to benchmark finite set of points algorithms [DPAM02, ZLT01, NDN<sup>+</sup>09, ZK04]. ZDT1 and ZDT2 can be used to assess the effect of concavity and convexity of the Pareto front on the different replacement strategies.

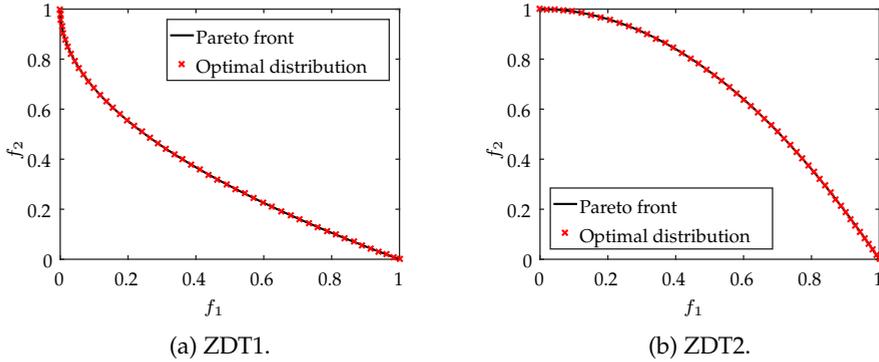


Figure 5.2.: Pareto fronts of the problems ZDT1 and ZDT2 including a 50-optimal distribution of points for  $\Psi(\mathbf{u}) := 1$ .

The DEB2DK and DO2DK problems feature mixed curvature Pareto fronts [BDDO04] (Figures 5.3 and 5.4). They both possess a parameter  $k$  that controls the number of convex bulges – also called knees – on the Pareto front. DO2DK additionally uses a parameter  $s$  that adds skewing to the front. Mixed curvature Pareto fronts usually possess a complex geometric shape potentially rendering finding

## 5. Preference-biased Pareto Front Approximations

a uniform finite set of points approximation more difficult.

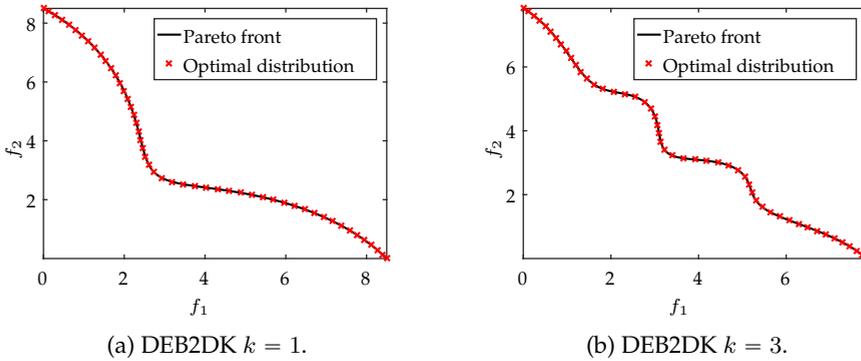


Figure 5.3.: Pareto fronts of the problem DEB2DK for  $k = 1$  and  $k = 3$  including a 50-optimal distribution of points for  $\Psi(\mathbf{u}) := 1$ .

The Pareto front of an instance of the Lamé problem is the part of a unit hypersphere in the positive orthant [ED07] (Figures 5.5 and 5.6). The parameter  $\gamma$  controls the curvature of the front. For any fixed  $\gamma$  the distance induced by the  $p$ -norm with  $p = \gamma$  of every Pareto optimal point to the origin is equal to 1. For example,  $\gamma = 2$  yields a circle and  $\gamma = 1$  a straight line as Pareto front. The values 0.25 and 4 for  $\gamma$  using two objectives and 0.5 for three objectives were chosen to assess the effect of extreme curvatures on the replacement strategies. As explained in Section 4.1, other optimal distribution concepts such as hypervolume are gravely affected by extreme curvatures making these problem instances interesting test cases.

The DTLZ family comprises a set of problems that is scalable in the number of objectives [DTLZ05]. Three objectives were chosen for all problem instances. DTLZ1's Pareto front is the unit simplex scaled by 0.5. Its inverted version [JD14] is considered as well. Hypervolume-based finite set of points approaches have as well shown to struggle in producing uniform approximations on the inverted DTLZ1 problem [IISN17]. DTLZ3's Pareto front is the part of the unit sphere in the positive orthant. The problem is known to be difficult to solve (Figure 5.7).

Finally, a new set of test problems is proposed: The B problem family consists of six three objective problems (Figures 5.8 and 5.9 and Definition 69). They appear simple in their mathematical structure compared to more recent problem families such as DTLZ and WFG [HHBW06], however the B family features problems that possess Pareto fronts with complicated shapes that have so far

### 5.3. Computational Analysis of Preference-biased Approximations

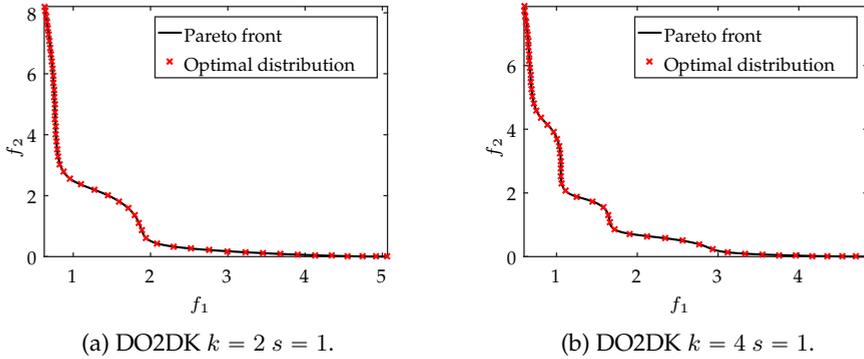


Figure 5.4.: Pareto fronts of the problem DO2DK for  $k = 2$  and  $k = 4$  with  $s = 1$  including a 50-optimal distribution of points for  $\Psi(\mathbf{u}) := 1$ . The optimal distribution is more dense towards the left extreme points, because the range of  $f_2$  among all Pareto points is larger compared to  $f_1$ . This effect can be prevented by normalizing objectives before distance computation.

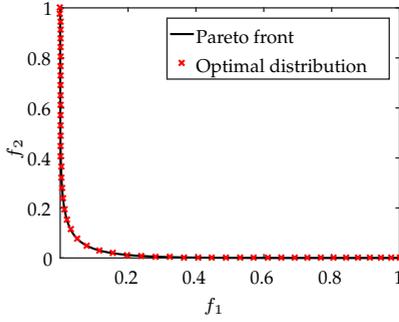
not been explored in MOO. Since the shape of the Pareto front may exert a large effect on the distribution of points on the front generated by a niching mechanism [IISN17], the B family is proposed and included in the study. B1's Pareto front is convex and formed like a spade, whereas the Pareto front of B2 is concave and shaped like a sun sail. The front of B3 resembles a linen that is folded towards the center at its four corners. B4's front is a plane. The front of B5 possesses a wave-like shape and B6's front resembles a smooth staircase. A list of all problems utilized in the study is found in Table 5.2. Their mathematical definitions are contained in Appendix B.

## 5. Preference-biased Pareto Front Approximations

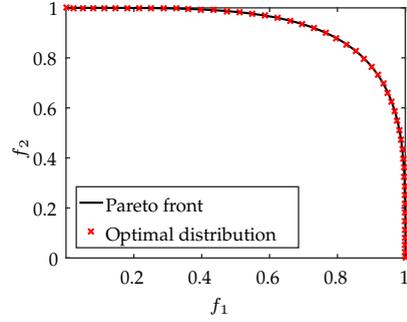
Table 5.2.: Problems utilized in the replacement strategy study. The table features the problem names including their parametrization, the source of publication, the number of objectives  $m$  and the number of decision variables  $n$ .

Name	Source	$m$	$n$
B1	new	3	12
B2	new	3	12
B3	new	3	12
B4	new	3	12
B5	new	3	12
B6	new	3	12
DEB2DK $k = 1$	[BDDO04]	2	30
DEB2DK $k = 3$	[BDDO04]	2	30
DO2DK $k = 2$ $s = 1$	[BDDO04]	2	30
DO2DK $k = 4$ $s = 1$	[BDDO04]	2	30
DTLZ1	[DTLZ05]	3	7
inverted DTLZ1	[JD14]	3	7
DTLZ3	[DTLZ05]	3	12
Lamé $\gamma = 0.25$	[ED07]	2	7
Lamé $\gamma = 0.5$	[ED07]	3	7
Lamé $\gamma = 4$	[ED07]	2	7
ZDT1	[ZDT00]	2	30
ZDT2	[ZDT00]	2	30

### 5.3. Computational Analysis of Preference-biased Approximations

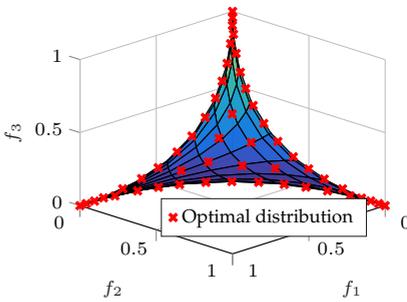


(a) Lamé  $\gamma = 0.25$ .

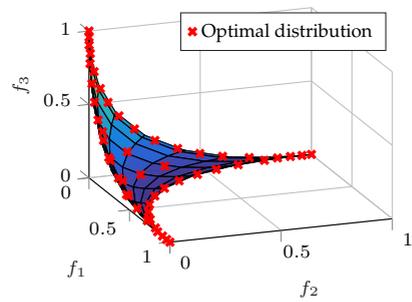


(b) Lamé  $\gamma = 4$ .

Figure 5.5.: Pareto fronts of the Lamé problem for two objectives and  $\gamma = 0.25$  and  $\gamma = 4$ , respectively.



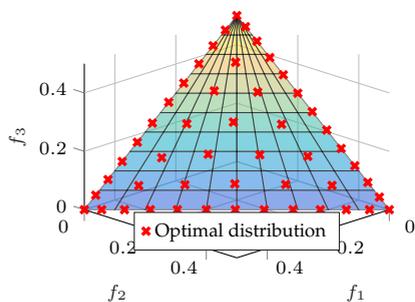
(a) Lamé  $\gamma = 0.5$  (front).



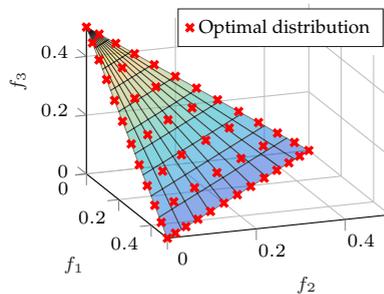
(b) Lamé  $\gamma = 0.5$  (side).

Figure 5.6.: Pareto fronts of the Lamé problem for three objectives with  $\gamma = 0.5$  from the front and side including a 50-optimal distribution of points for  $\Psi(\mathbf{u}) := 1$ .

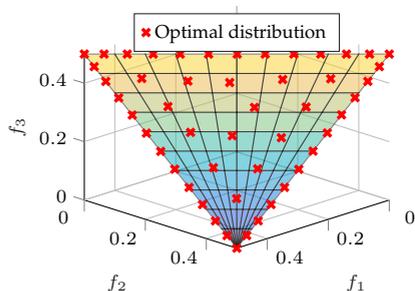
## 5. Preference-biased Pareto Front Approximations



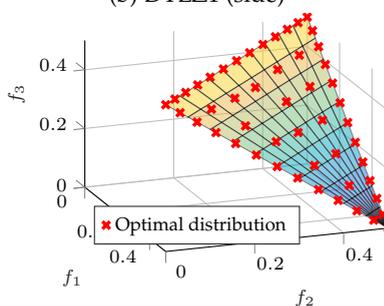
(a) DTLZ1 (front)



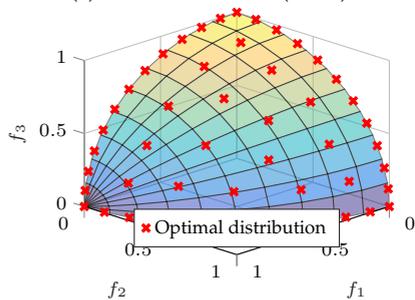
(b) DTLZ1 (side)



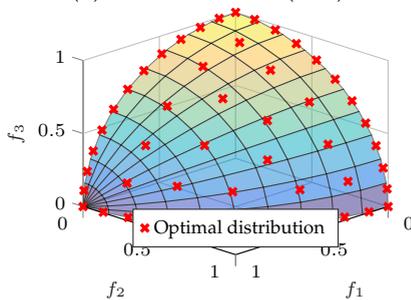
(c) Inverted DTLZ1 (front).



(d) Inverted DTLZ1 (side).



(e) DTLZ3 (front)



(f) DTLZ3 (side)

Figure 5.7.: Pareto fronts of the problems DTLZ1, inverted DTLZ1 and DTLZ3 from the front and the side including a 50-optimal distribution of points for  $\Psi(\mathbf{u}) := 1$ .

### 5.3. Computational Analysis of Preference-biased Approximations

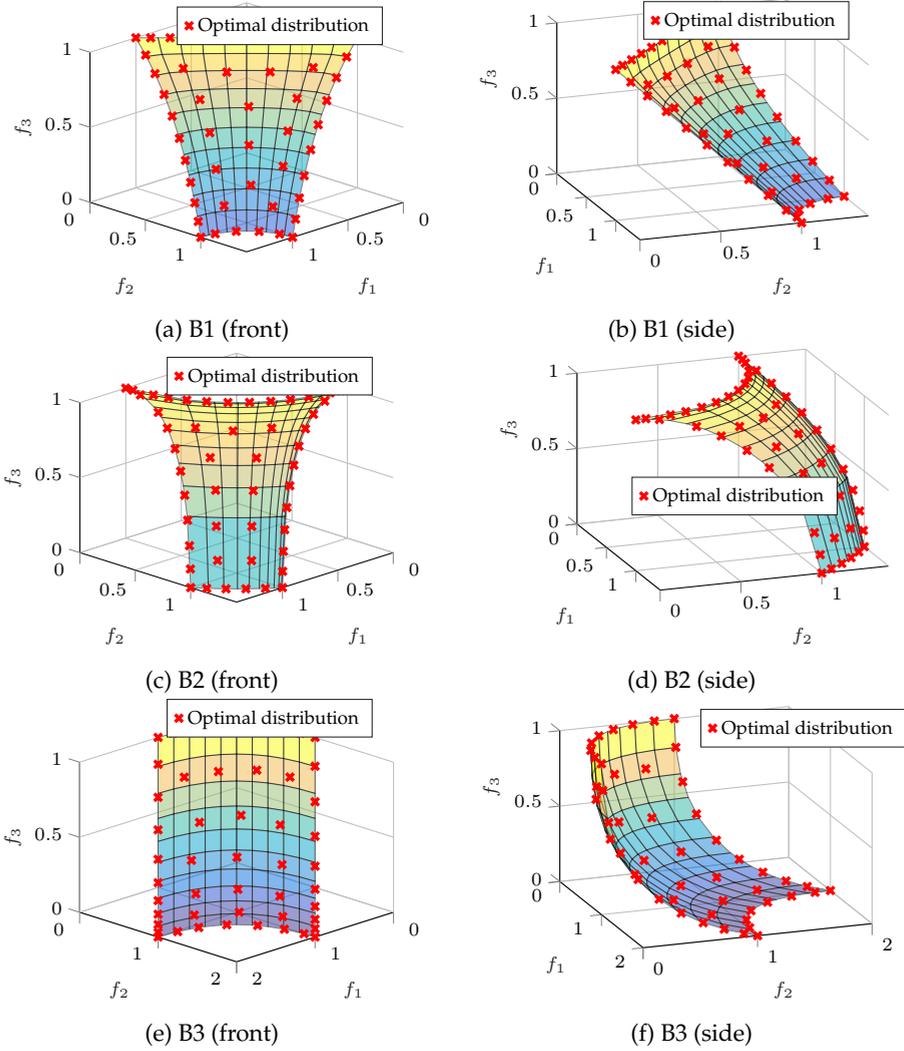


Figure 5.8.: Pareto fronts of the problems B1, B2 and B3 from the front and the side including a 50-optimal distribution of points for  $\Psi(\mathbf{u}) := 1$ .

## 5. Preference-biased Pareto Front Approximations

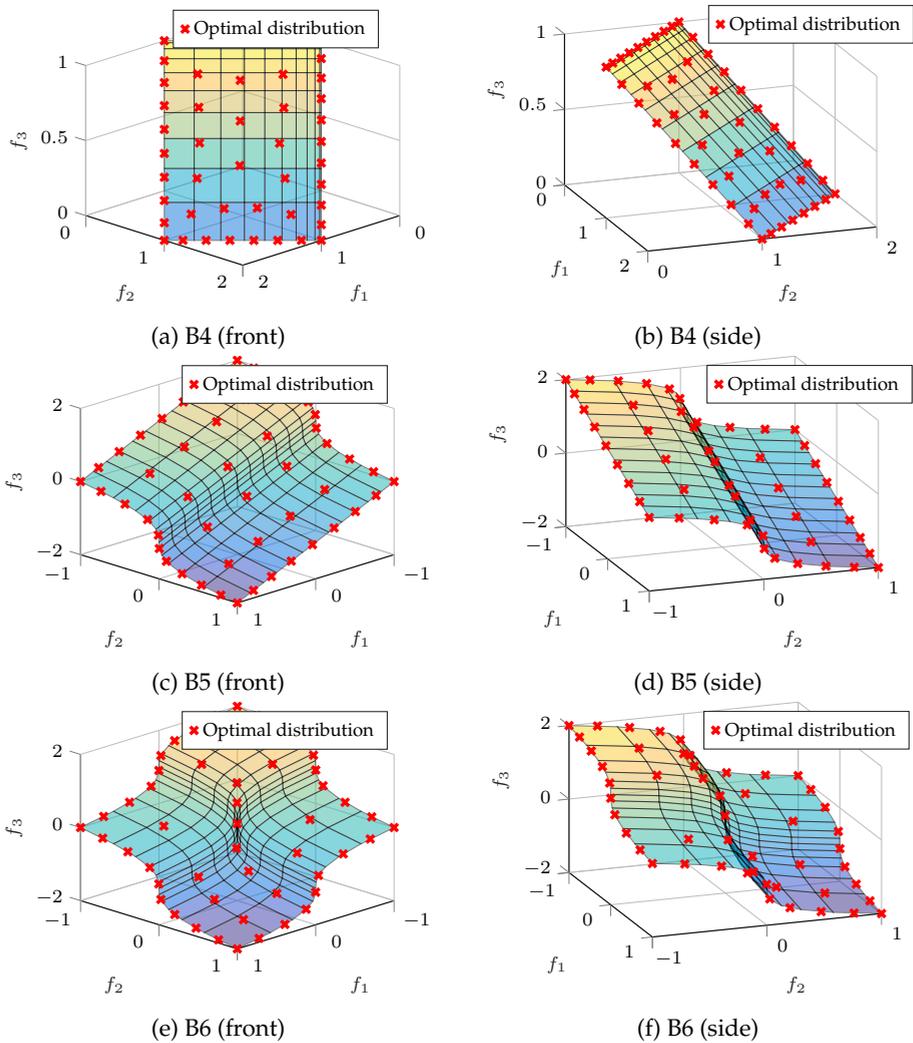


Figure 5.9.: Pareto fronts of the problems B4, B5 and B6 from the front and the side including a 50-optimal distribution of points for  $\Psi(\mathbf{u}) := 1$ .

### 5.3.1.3. Replacement Strategies Study Performance Indicators

Two performance indicators are used to evaluate the simulation results. The first indicator measures the energy as of Equation (5.1) of the approximation  $S$  generated by ESPEA relative to the energy of a reference  $N$ -optimal distribution of points  $P$ . The indicator relative energy (RE) is computed by dividing the energy of the approximation by the energy of the reference set (Definition 56).

**Definition 56** (relative energy). *Let  $S, P \subset \mathbb{R}^m$  be finite. The relative energy (RE)  $RE(S, P)$  is defined as*

$$RE(S, P) = \frac{U(S)}{U(P)}. \quad (5.15)$$

The reference sets were generated by formulating mathematical programs to find the minimum of Equation (5.1) for each test problem. To simplify the search for the energy minimum, parametric equations for the Pareto fronts of all presented test problems were computed. Thereby, Pareto dominated points are eliminated from the search space. For any of the presented problems, it holds that all objective functions are non-conflicting in decision variables  $\{x_m, \dots, x_n\}$ . This means that for a given problem, all these variables are required to take on a specific value to generate a Pareto optimal point.<sup>4</sup> Any deviation from this value leads the resulting objective vector to move away from the front. The parametric equations retain only  $\{x_1\}$  as free variable for two objectives and  $\{x_1, x_2\}$  as free variables for three variables. Any feasible value for a free variable produces a Pareto optimal point. The MATLAB software using the solver *fmincon* was used in combination with the *interior-point* option to generate the reference fronts. A maximum number of 50 000 function evaluations was chosen as stopping criterion.

RE is an indicator for measuring the uniformity of a finite set of points Pareto front approximation if  $\Psi(\mathbf{u}) := 1$  is used as scalarization function. Smaller values of the RE indicator signalize a higher quality of approximation. RE is only a useful means of comparison if  $S$  and  $P$  have the same size and all elements of both sets are located on the Pareto front. Evidently, any  $N - 1$ -optimal distribution possesses less energy than an  $N$ -optimal distribution, since fewer addends enter Equation (5.1). In the non-preference case, all scalarization values are set to 1. Distributing points across the entire feasible objective space yields a smaller RE compared to focusing all  $N$  points on the Pareto front. Therefore, the search space was restricted to the Pareto optimal set  $X_p$  in the study that uses RE as performance indicator.

<sup>4</sup>For all B-problems for example, decision variables  $\{x_m, \dots, x_n\}$  need to take on the value 0 to generate Pareto optimal points.

## 5. Preference-biased Pareto Front Approximations

The second performance indicator that is used to compare the simulation results is inverted generational distance (IGD). IGD measures the distance between two sets of objective vectors. Given two sets  $S$  and  $P$ , IGD computes the distance of every point in  $P$  to its closest neighbor in  $S$ . The mean of all these distances constitutes the IGD (Definition 57).

**Definition 57** (inverted generational distance [VVL98]). *Let  $S, P \subset \mathbb{R}^m$  be finite. The inverted generational distance (IGD)  $\text{IGD}(S, P)$  is defined as*

$$\text{IGD}(S, P) = \frac{1}{|P|} \sum_{\mathbf{u} \in P} \min_{\mathbf{v} \in S} \|\mathbf{u} - \mathbf{v}\|_2. \quad (5.16)$$

IGD is commonly used in MOO to assess the quality of finite set of points approximations of the entire front. In this case,  $P$  is a uniform, finite, discretized subset of the Pareto front and  $S$  is the approximation generated by an MOEA. IGD measures both how close the approximation is to the true front and how well spread the points are across the entire front. In evolutionary MOO the former property is denoted by convergence and the latter by diversity.

In the computational study that is presented in this section,  $P$  is chosen as the  $N$ -optimal distribution of points. IGD thereby reveals how close the approximation generated by ESPEA is to the  $N$ -optimal distribution. The simulation in which IGD was measured to compare the replacement strategies retained the original search spaces of all problems in the study. There were no restrictions in place that limited the search space to the Pareto optimal set. Thereby, the convergence of ESPEA's population towards the Pareto front using the different replacement strategies can be assessed as well. Smaller IGDs imply that the approximate distribution is closer to the optimal distribution in the objective space.

Although RE and IGD measure both how well a set of points approximates an  $N$ -optimal distribution of points, they differ in their explanatory power. Any  $N$ -sized set of Pareto optimal points  $S$ , whose energy is close to that of the  $N$ -optimal distribution  $P$ , would constitute a close-to-optimal distribution in terms of RE. The elements of set  $S$ , however, may be located at different positions than those of  $P$  (see Figure 5.10), which result in a larger IGD. IGD measures only the distance between  $S$  and  $P$  in the objective space and makes no statement about the quality of distribution in terms of RE. On the other hand, RE cannot be used to quantify the convergence of ESPEA towards the Pareto front if the entire feasible set  $X$  is considered, since few to none archive members are expected to be located exactly on the Pareto front.

### 5.3. Computational Analysis of Preference-biased Approximations

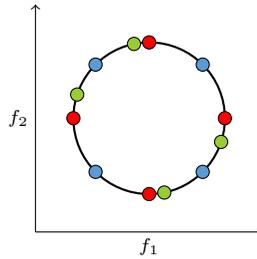


Figure 5.10.: Consider the  $f_1$ - $f_2$ -plot of a three objective Pareto front that consists of a circle and the three sets of red ( $R$ ), blue ( $B$ ) and green ( $G$ ) points. The sets  $R$  and  $B$  constitute a 4-optimal distribution. If  $R$  is chosen as reference set, then  $RE(B, R) = 1 < RE(G, R)$ , however  $IGD(B, R) > IGD(G, R)$ .

#### 5.3.1.4. Discussion of the Replacement Strategies Study Results

EAs utilize random numbers in generating new candidate solutions. Each run of an EA may yield a different outcome even if the same start population is used. EAs must therefore be run multiple times on the same problem to allow the derivation of meaningful statements about their performance. ESPEA was run 100 times with each replacement strategy on every problem. In each run, 50 000 function evaluations were chosen as termination criterion.

The development of RE across the 50 000 function evaluations is depicted in Figures 5.11 to 5.13. The results show a near universal tendency. The replacement strategies WIN and LED clearly outperform BFP in early iterations. Although all three strategies start with the same initial population, WIN and LED converge faster towards the energy optimum. The performance range of BFP is also larger compared to the other two methods. WIN and LED appear to exhibit a near identical performance on all problems. This observation suggests that both strategies mostly select the same archive members for replacement.

The results also show that focusing on the elimination of the worst archive members (LED and WIN) is more beneficial compared to inserting new individuals such that their effect on overall energy is minimized BFP. This observation caters to the Darwinian notion of survival of the fittest. All three replacement strategies, however, arrive at a close approximation of the energy optimum within 1000 function evaluations. This allows the conclusion that BFP, WIN and LED are all suitable replacement strategies for ESPEA.

In order to evaluate the performance differences between the three replacement strategies in later iterations, RE after 50 000 function evaluations is reported in Table 5.3. The performance differences observed in early iterations continue to

## 5. Preference-biased Pareto Front Approximations

Table 5.3.: Replacement strategies study – RE. Median and IQR (as subscript) results after 50 000 function evaluations. Only significant digits are reported after omitting the leading 1. Negative values imply that ESPEA outperformed the reference set and RE is smaller than 1. Best performance is colored in green, second best in blue. WIN and LED outperform BFP.

	BFP	WIN	LED
B1	-1.37e-3 <sub>8.7e-4</sub>	-1.29e-3 <sub>1.0e-3</sub>	-1.40e-3 <sub>9.6e-4</sub>
B2	-1.24e-3 <sub>6.5e-4</sub>	-1.09e-3 <sub>7.0e-4</sub>	-1.21e-3 <sub>7.6e-4</sub>
B3	1.41e-4 <sub>6.3e-4</sub>	4.59e-4 <sub>4.7e-4</sub>	9.47e-5 <sub>7.2e-4</sub>
B4	-1.07e-3 <sub>8.4e-4</sub>	-1.12e-3 <sub>7.6e-4</sub>	-1.20e-3 <sub>7.9e-4</sub>
B5	-1.38e-4 <sub>6.7e-4</sub>	-3.68e-4 <sub>1.1e-3</sub>	-4.39e-4 <sub>9.5e-4</sub>
B6	-2.16e-3 <sub>1.1e-3</sub>	-2.24e-3 <sub>1.2e-3</sub>	-2.23e-3 <sub>7.8e-4</sub>
DEB2DK $k = 1$	5.44e-4 <sub>5.1e-4</sub>	4.90e-4 <sub>5.3e-4</sub>	5.14e-4 <sub>5.2e-4</sub>
DEB2DK $k = 3$	5.56e-4 <sub>5.4e-4</sub>	4.62e-4 <sub>5.5e-4</sub>	5.11e-4 <sub>4.5e-4</sub>
DO2DK $k = 2$ $s = 1$	5.77e-4 <sub>5.1e-4</sub>	4.72e-4 <sub>5.2e-4</sub>	4.02e-4 <sub>4.6e-4</sub>
DO2DK $k = 4$ $s = 1$	6.98e-4 <sub>5.5e-4</sub>	5.25e-4 <sub>5.4e-4</sub>	5.75e-4 <sub>5.9e-4</sub>
DTLZ1	-2.18e-3 <sub>9.5e-4</sub>	-2.51e-3 <sub>6.9e-4</sub>	-2.36e-3 <sub>7.3e-4</sub>
invDTLZ1	-7.75e-4 <sub>1.0e-3</sub>	-8.00e-4 <sub>8.3e-4</sub>	-7.41e-4 <sub>7.9e-4</sub>
DTLZ3	1.27e-3 <sub>1.2e-3</sub>	8.72e-4 <sub>1.2e-3</sub>	8.50e-4 <sub>1.2e-3</sub>
Lame $m = 2$ $\gamma = 0.25$	5.02e-4 <sub>5.6e-4</sub>	3.18e-4 <sub>5.0e-4</sub>	3.82e-4 <sub>4.3e-4</sub>
Lame $m = 3$ $\gamma = 0.5$	1.11e-4 <sub>4.5e-4</sub>	7.96e-5 <sub>4.5e-4</sub>	-9.14e-5 <sub>5.1e-4</sub>
Lame $m = 2$ $\gamma = 4$	7.06e-4 <sub>6.4e-4</sub>	6.71e-4 <sub>6.0e-4</sub>	5.52e-4 <sub>4.7e-4</sub>
ZDT1	6.09e-4 <sub>4.8e-4</sub>	4.20e-4 <sub>5.7e-4</sub>	4.91e-4 <sub>5.0e-4</sub>
ZDT2	5.16e-4 <sub>7.2e-4</sub>	4.25e-4 <sub>4.6e-4</sub>	3.67e-4 <sub>3.8e-4</sub>

manifest themselves at the end of the execution. WIN and LED clearly outperform BFP on nearly all test problems. There exist only three problems for which BFP achieves a smaller RE compared to any of the other two methods. On these problems, the range of RE among all three strategies is small suggesting that the actual performance difference is marginal.

The values listed in Table 5.3 confirm the observations of the convergence plots that the three replacement strategies are able to find a close approximation of the energy minimum. In some cases, they are even able to outperform the benchmark obtained by solving a corresponding mathematical program. This observation confirms that the EA is a suitable approach for finding energy minima.

Performance differences after 50 000 function evaluations can be further explored by assessing the distribution of RE achieved by the different replacement strategies across the 100 runs. Figure 5.14 shows boxplots of RE for the individual strategies after 50 000 function evaluations. As in Table 5.3, only significant digits are displayed. Outliers were removed for better visualization. The boxplots reveal that there exist only marginal performance differences between the three replacement strategies. Although the first quartile of LED and WIN is on average smaller than BFP's, the IQRs of all methods largely overlap.

### 5.3. Computational Analysis of Preference-biased Approximations

A statistical analysis is performed to check the performance differences for significance. Statistical hypothesis tests allow drawing conclusions about whether differences observed in data samples are systemic or merely coincidental. A statistical test consists of a null and alternative hypothesis, where both are falsifiable statements and the alternative is the exact opposite of the null hypothesis. The null hypothesis is assumed to be true unless there exists sufficient evidence that suggests otherwise. Evidence is generated by computing a test statistic, which is a numerical summary of the data samples. The test statistic is compared against the expected sample distribution, whereas the sample distribution must either be known or inferred before testing. The comparison yields the  $p$ -value, which states the probability of observing the sample data or a more extreme outcome assuming that the null hypothesis is true. If the  $p$ -value falls below a certain threshold the null hypothesis is rejected and the observed difference between the samples is deemed significant. Equivalently, it can be stated that a difference is observed with confidence [FKPT12].

The first step of the statistical analysis consists of inferring the sample distribution. Hypothesis testing mainly distinguishes between parametric and non-parametric tests. Parametric tests possess a greater statistical power implying that they are more likely to discover a significant effect if one exists. Parametric tests require that the sample data follows a specific distribution. According to the central limit theorem, the sum of independent and identical distributed variables follows asymptotically a normal distribution [FKPT12]. Since each algorithm run on a given test problem is independent of previous and subsequent runs, performance indicator values of individual runs are assumed to be independent. Running a given algorithm on the same test problem using identical configurations should result in indicator values that stem from the same distribution. Therefore, results for RE were first checked for normal distribution.

Each sample of 100 runs for every algorithm and problem was tested for normal distribution using the Anderson-Darling test [AD52]. The null hypothesis of the Anderson-Darling test states that the sample follows a normal distribution. For every test problem the null hypothesis of normal distributed RE was rejected for at least one algorithm at a significance level of 0.05 (Table 5.4).

Therefore, the non-parametric Kruskal-Wallis [KW52] test was used to assess, whether the observed performance differences are significant. The null hypothesis of the Kruskal-Wallis states that all considered samples – RE grouped by the three replacement strategies on a given test problem – stem from the same distribution, whereas no assumptions about the shape of the distribution is made. If the simulation results suggest that all samples come from the same distribution, the observed performance differences are not significant. The  $p$ -values of the Kruskal-Wallis test are reported in Table 5.4. Out of the 18 consid-

## 5. Preference-biased Pareto Front Approximations

ered test problems, significant performance differences are only observed on six of them.

Table 5.4.: Replacement strategies study – RE. The table reports  $p$ -values of the Anderson-Darling and ANOVA/Kruskal-Wallis test. Significant results at a 5 % level are highlighted in green. Significant difference are only reported on six out of 18 problem instances.

	Anderson-Darling			Kruskal-Wallis
	BFP	WIN	LED	
B1	0.0005	0.0005	0.0005	0.5062
B2	0.0005	0.0005	0.0005	0.9051
B3	0.0005	0.0005	0.0005	0.0491
B4	0.0016	0.0005	0.0005	0.2636
B5	0.1732	0.0108	0.2134	0.2663
B6	0.0005	0.0005	0.0005	0.2701
DEB2DK $k = 1$	0.0005	0.0005	0.0005	0.2277
DEB2DK $k = 3$	0.0005	0.0005	0.0005	0.1163
DO2DK $k = 2$ $s = 1$	0.0005	0.0005	0.0005	0.0120
DO2DK $k = 4$ $s = 1$	0.0005	0.0005	0.0005	0.1732
DTLZ1	0.0021	0.0005	0.0005	0.1229
invDTLZ1	0.0005	0.0005	0.0005	0.8089
DTLZ3	0.0005	0.0005	0.0005	0.1079
Lamé $m = 2$ $\gamma = 0.25$	0.0005	0.0005	0.0005	0.0005
Lamé $m = 3$ $\gamma = 0.5$	0.0293	0.0005	0.0005	0.0485
Lamé $m = 2$ $\gamma = 4$	0.0009	0.0005	0.0087	0.0366
ZDT1	0.0005	0.0005	0.0005	0.0911
ZDT2	0.0005	0.0005	0.0005	0.0215

A rejection of the null hypothesis of the Kruskal-Wallis test only indicates that the median performance of at least one replacement strategy significantly differs from those of the other strategies. In order to assess which strategies differ from each other, a post-hoc analysis is required that performs pairwise comparisons between the different strategies [FKPT12]. Dunn’s method [Dun64] was used for the post-hoc analysis.

The results of the post-hoc analysis are displayed in Table 5.5. BFP is significantly outperformed by WIN on two and by LED on four test problems. No significant performance differences between WIN and LED were reported. A significant performance difference cannot be confirmed by the post-hoc analysis on the three objective Lamé problem. This is due to the  $p$ -value reported by the Kruskal-Wallis test being very close to 0.05 (Table 5.4) implying that the observed difference is not very strong. The results in Tables 5.4 and 5.5 do not suggest a systemic performance difference between the three methods. It is worth noticing though that the Pareto fronts of all problems, on which significant performance differences were detected, exhibit either strong convex or concave curvatures. This suggests that WIN and LED outperform BFP on prob-

### 5.3. Computational Analysis of Preference-biased Approximations

lems, whose curvature is strongly pronounced.

Table 5.5.: Replacement strategies study – RE. The table reports  $p$ -values of a post-hoc analysis. Each column lists the  $p$ -value of the comparison of BFP with respect to WIN and LED on the given problem. Significant results at a 5 % level are highlighted in green for smallest and blue for second smallest  $p$ -value. More significant performance differences are reported for LED.

	WIN	LED
B3	0.0430	0.6706
DO2DK $k = 2$ $s = 1$	0.1755	0.0099
Lamé $m = 2$ $\gamma = 0.25$	0.0008	0.0076
Lamé $m = 3$ $\gamma = 0.5$	0.9999	0.0906
Lamé $m = 2$ $\gamma = 4$	0.6954	0.0321
ZDT2	0.1258	0.0236

Summarizing the results for RE, WIN and LED outperform BFP in early iterations of the search. Although performance differences persevere till later iterations, only a few statistically significant differences can be reported. Still, WIN and LED appear to perform evenly well, whereas the use of the BFP strategy is discouraged.

5. Preference-biased Pareto Front Approximations

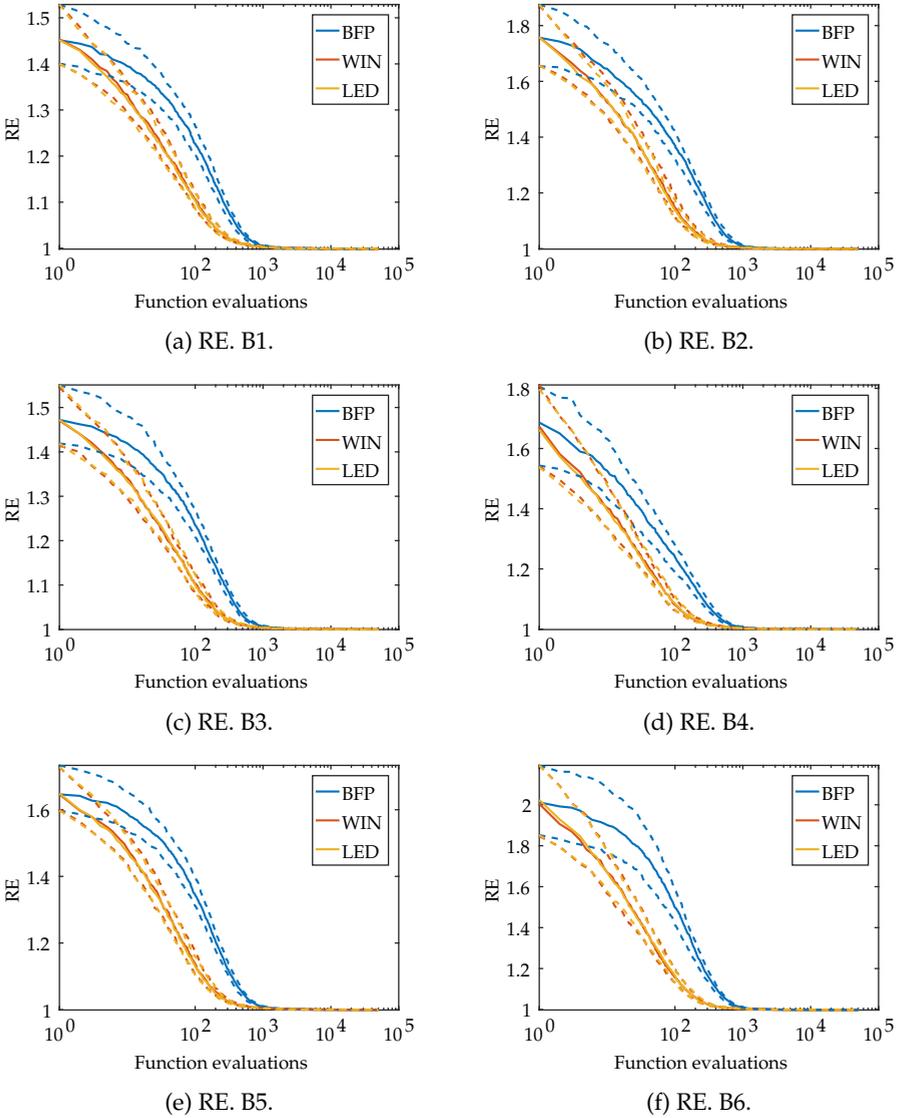


Figure 5.11.: Replacement strategies study – convergence of RE for problems B1 to B6. Straight lines indicate median performance and dashed lines inter-quartile ranges (IQRs) of performance.

### 5.3. Computational Analysis of Preference-biased Approximations

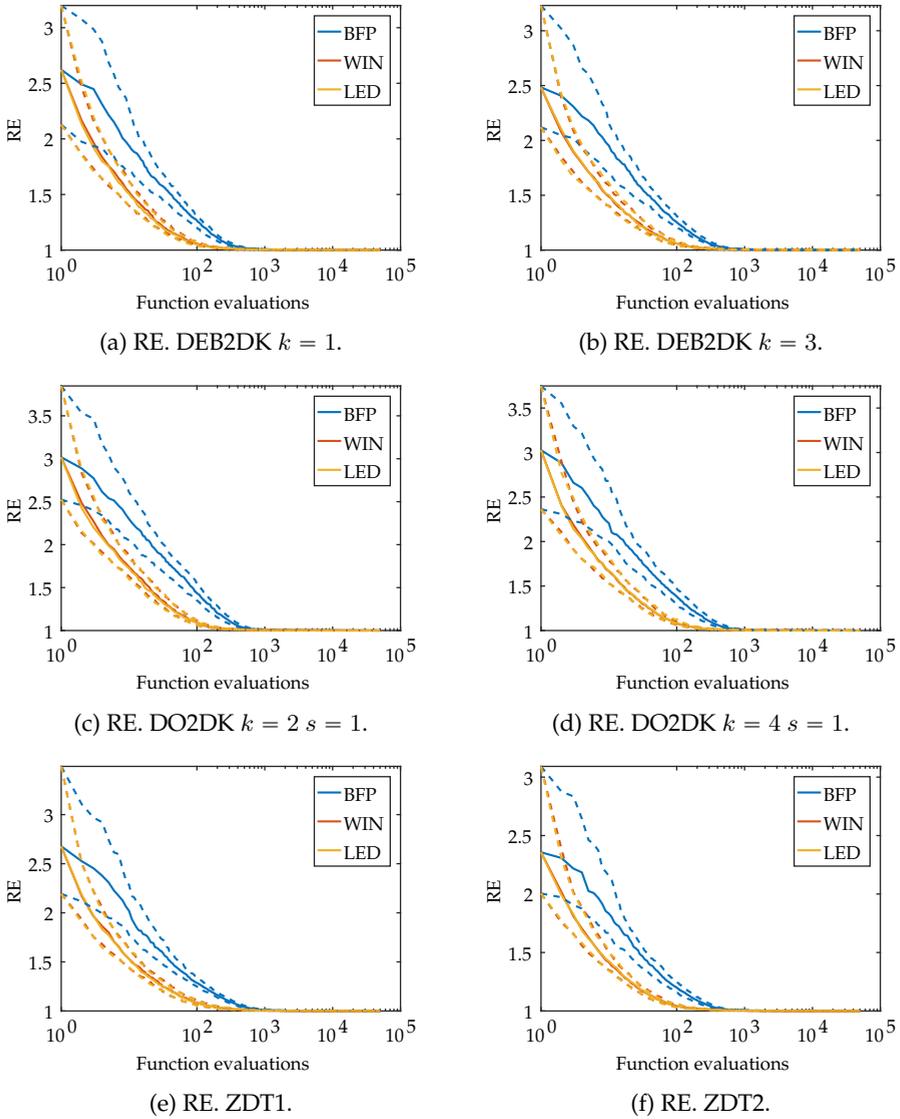


Figure 5.12.: Replacement strategies study – convergence of RE for problems DEB2DK, DO2DK, ZDT1 and ZDT2. Straight lines indicate median performance and dashed lines IQRs of performance.

## 5. Preference-biased Pareto Front Approximations

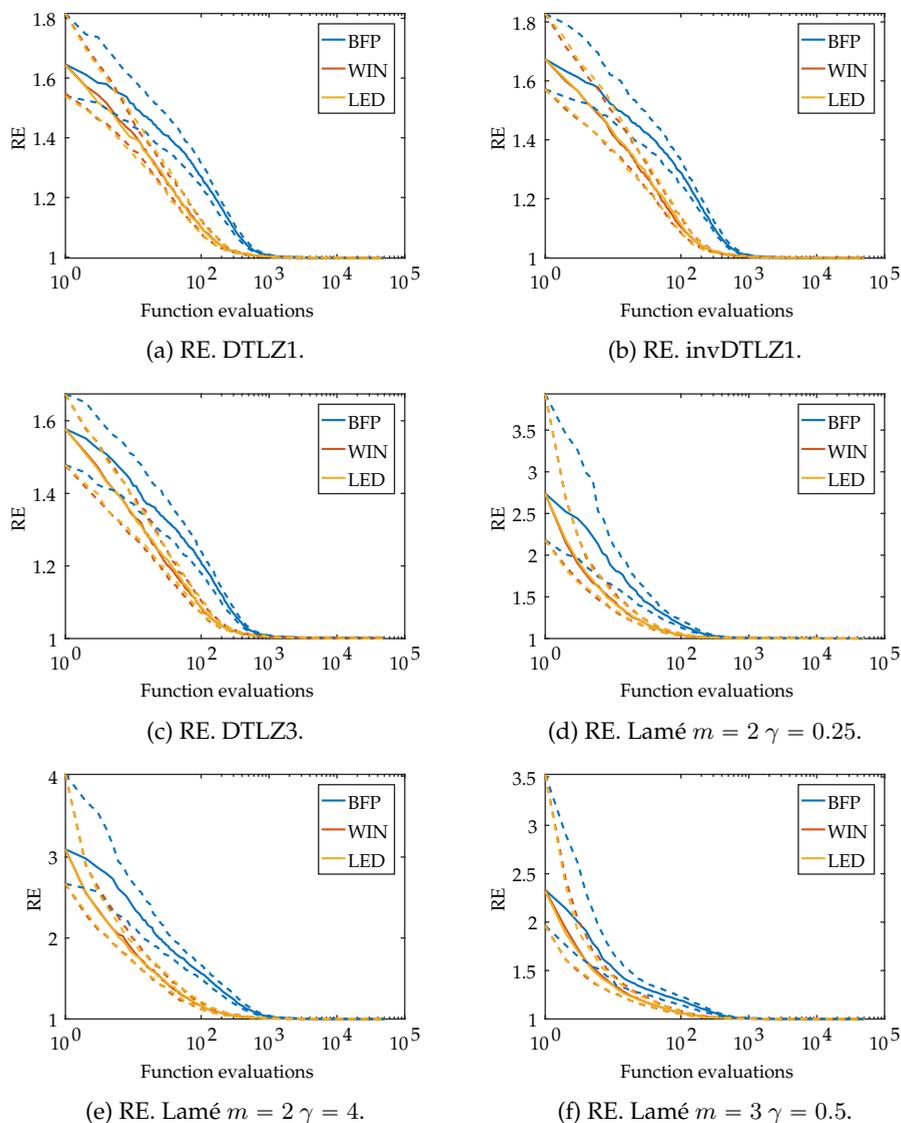


Figure 5.13.: Replacement strategies study – convergence of RE for problems DTLZ1, invDTLZ1, DTLZ3 and the Lamé problems. Straight lines indicate median performance and dashed lines IQRs of performance.

### 5.3. Computational Analysis of Preference-biased Approximations

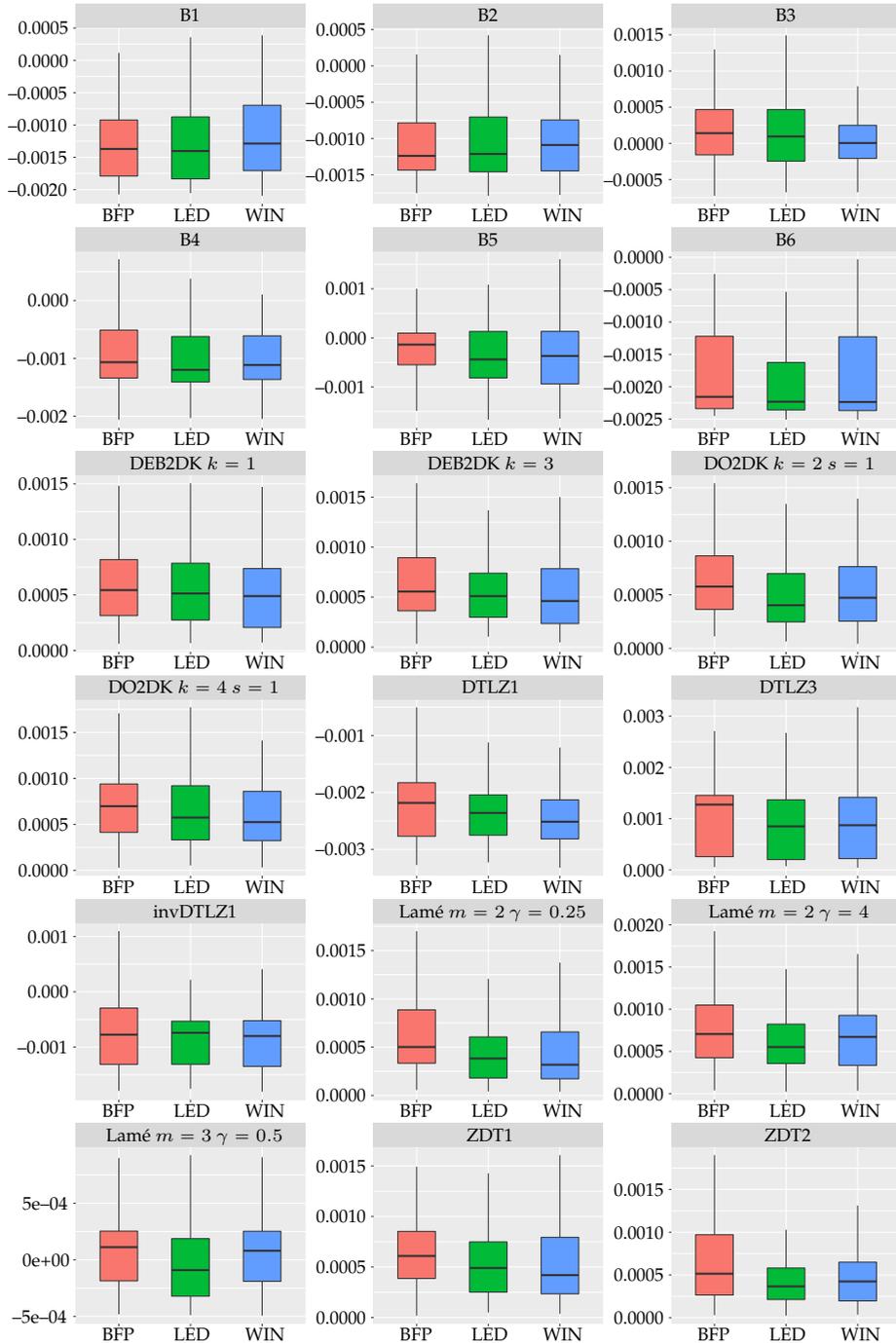


Figure 5.14.: Replacement strategies study – boxplots of RE after 50 000 function evaluations.

## 5. Preference-biased Pareto Front Approximations

Next, the three replacement strategies are compared with respect to the IGD performance indicator. Figures 5.15 to 5.17 show the development of IGD across 50 000 function evaluations for all test problems. The analysis focuses on late iterations, since the initial populations are expected to be far away from the Pareto front. The results indicate a rather small performance difference between the three replacement strategies. Notable differences are only observable on B1, B3, B4 and DO2DK ( $k = 4$ ), on which WIN and LED once again outperform BFP.

On nearly all problem instances, all three replacement strategies show early convergence towards the energy minimum between 10 000 and 20 000 function evaluations. This is a common range for MOEAs that compute finite set of points approximations [BNE07] and thereby suggests that ESPEA is a suitable algorithm for obtaining finite set of points approximations as well. A notable exception is DTLZ3, for which convergence is only achieved between 30 000 and 40 000 function evaluations. This reflects the observation that DTLZ3 is known to be difficult to solve [TF07]. Still, all three replacement strategies perform well with respect to IGD.

The performance of ESPEA slightly deteriorates in late iterations on DEB2DK ( $k = 1$ ) and ZDT2. One explanation for this behavior could be that ESPEA converges towards a local energy optimum, whose energy differs marginally from the global optimum, but which is further away in the objective space from the global optimum (see Table 5.3 and Figure 5.10).

IGD after 50 000 function evaluations is displayed in Table 5.6. The figures show that WIN and LED once again outperform BFP on most test problems, although BFP achieves better results on more test problems compared to RE. The magnitude of the performance difference between all three replacement strategies – the effect size, however, is small. This observation suggests that all three strategies exhibit negligible performance differences if sufficient function evaluations are performed.

The distribution of IGD after 50 000 function evaluations is shown in Figure 5.18. Outliers were again removed for a better visualizations. Similar to RE, the IQRs of the three replacement strategies tend to overlap on most problems, although BFP appears to perform slightly worse on average. On B1, B2 and B3, however, BFP is clearly outperformed by the other two strategies. This observation is already hinted at in the convergence plots (Figure 5.15). A post-hoc analysis is performed to verify, whether the observed differences are significant.

The post-hoc analysis was performed in the same manner as it was done for RE (Table 5.7). On six problem instances, the Anderson-Darling test did not reject the null hypothesis of normally distributed data for all three replacement strategies. For these six problems, IGD was tested for variance homogeneity

### 5.3. Computational Analysis of Preference-biased Approximations

Table 5.6.: Replacement strategies study – IGD. Median and IQR (as subscript) results after 50 000 function evaluations. Best performances are colored in green, second best in blue. WIN and LED are the top performing replacement strategies.

	BFP	WIN	LED
B1	8.34e-3 <sub>8.4e-4</sub>	7.66e-3 <sub>8.0e-4</sub>	7.54e-3 <sub>7.6e-4</sub>
B2	7.44e-3 <sub>7.7e-4</sub>	7.21e-3 <sub>7.4e-4</sub>	7.19e-3 <sub>8.8e-4</sub>
B3	1.11e-2 <sub>1.1e-3</sub>	1.00e-2 <sub>1.0e-3</sub>	9.72e-3 <sub>1.0e-3</sub>
B4	1.06e-2 <sub>9.7e-4</sub>	9.60e-3 <sub>1.0e-3</sub>	9.69e-3 <sub>9.0e-4</sub>
B5	7.89e-3 <sub>1.1e-3</sub>	7.75e-3 <sub>1.3e-3</sub>	7.76e-3 <sub>9.0e-4</sub>
B6	7.80e-3 <sub>9.4e-4</sub>	7.67e-3 <sub>9.9e-4</sub>	7.62e-3 <sub>1.1e-3</sub>
DEB2DK $k = 1$	1.30e-3 <sub>2.3e-4</sub>	1.31e-3 <sub>2.2e-4</sub>	1.31e-3 <sub>2.4e-4</sub>
DEB2DK $k = 3$	1.25e-3 <sub>3.8e-4</sub>	1.24e-3 <sub>4.2e-4</sub>	1.21e-3 <sub>4.2e-4</sub>
DO2DK $k = 2$ $s = 1$	1.38e-3 <sub>3.9e-4</sub>	1.35e-3 <sub>3.2e-4</sub>	1.38e-3 <sub>3.9e-4</sub>
DO2DK $k = 4$ $s = 1$	1.42e-3 <sub>4.3e-4</sub>	1.26e-3 <sub>3.6e-4</sub>	1.33e-3 <sub>3.5e-4</sub>
DTLZ1	7.53e-3 <sub>1.1e-3</sub>	7.62e-3 <sub>1.2e-3</sub>	7.57e-3 <sub>1.0e-3</sub>
invDTLZ1	6.37e-3 <sub>9.2e-4</sub>	6.45e-3 <sub>1.3e-3</sub>	6.37e-3 <sub>1.2e-3</sub>
DTLZ3	9.61e-3 <sub>1.2e-3</sub>	9.02e-3 <sub>1.9e-3</sub>	9.15e-3 <sub>1.6e-3</sub>
Lamé $m = 2$ $\gamma = 0.25$	1.57e-3 <sub>3.0e-4</sub>	1.59e-3 <sub>4.2e-4</sub>	1.53e-3 <sub>3.4e-4</sub>
Lamé $m = 3$ $\gamma = 0.5$	5.08e-3 <sub>6.0e-4</sub>	4.94e-3 <sub>7.4e-4</sub>	4.98e-3 <sub>6.3e-4</sub>
Lamé $m = 2$ $\gamma = 4$	1.46e-3 <sub>3.8e-4</sub>	1.44e-3 <sub>5.5e-4</sub>	1.45e-3 <sub>3.3e-4</sub>
ZDT1	1.24e-3 <sub>2.0e-4</sub>	1.21e-3 <sub>2.6e-4</sub>	1.25e-3 <sub>3.1e-4</sub>
ZDT2	1.15e-3 <sub>4.0e-4</sub>	1.09e-3 <sub>4.0e-4</sub>	1.10e-3 <sub>3.8e-4</sub>

using Bartlett’s test [Bar37]. The null hypothesis of the Bartlett test states that all samples come from distributions with equal variances – a property denoted by variance homogeneity or homoscedasticity. Homoscedasticity is a necessary requirement for conducting hypothesis tests that assume normally distributed data. In all six cases, variance homogeneity was not rejected allowing the application of an ANOVA test. The null hypothesis of ANOVA states that all samples come from the same normal distribution [FKPT12]. If the null hypothesis was rejected, Dunn’s method was used as well for the post-hoc comparison.

The post-hoc analysis reveals that there exist six problems on which significant performance differences occur (Table 5.8). On five of these problems, BFP is significantly outperformed by both WIN and LED. Performance differences between WIN and LED were not found to be significant. Comparing Tables 5.5 and 5.8 shows that RE and IGD detect significant performance differences mostly on different problems. Significant differences are only observed on B3 for both indicators. This further illustrates the fact that RE and IGD measure different aspects of the quality of approximation.

Taking all results presented in this section into account, a clear recommendation for utilizing either the WIN or LED strategy instead of BFP can be given. The performance difference between WIN and LED is not significant. Still, applying WIN yields better performance indicator values on most test problems, which is why WIN is chosen as replacement strategy in all subsequent studies.

## 5. Preference-biased Pareto Front Approximations

Table 5.7.: Replacement strategies study – IGD. The table shows  $p$ -values of the Anderson-Darling and ANOVA/Kruskal-Wallis tests. Significant results at a 5 % level for the ANOVA/Kruskal-Wallis test are highlighted in green. Significant performance differences are only observed on six out of 18 test problems.

	Anderson-Darling			Bartlett	ANOVA/Kruskal-Wallis
	BFP	WIN	LED		
B1	0.6383	0.8718	0.1987	0.2317	0.0000
B2	0.3318	0.2866	0.7997	0.7159	0.0000
B3	0.3921	0.3150	0.7208	0.5210	0.0000
B4	0.0580	0.9823	0.8606	0.3430	0.0000
B5	0.5666	0.1271	0.4816	0.7955	0.5968
B6	0.8479	0.3183	0.0445	n/a	0.0802
DEB2DK $k = 1$	0.1758	0.0167	0.0967	n/a	0.7090
DEB2DK $k = 3$	0.0005	0.0120	0.0016	n/a	0.2668
DO2DK $k = 2$ $s = 1$	0.5802	0.0035	0.0512	n/a	0.3625
DO2DK $k = 4$ $s = 1$	0.1958	0.9298	0.6535	0.5501	0.0010
DTLZ1	0.2252	0.0005	0.0549	n/a	0.9925
invDTLZ1	0.2810	0.1919	0.0051	n/a	0.8102
DTLZ3	0.0005	0.0005	0.0005	n/a	0.0003
Lame $m = 2$ $\gamma = 0.25$	0.3053	0.0005	0.0005	n/a	0.3690
Lame $m = 3$ $\gamma = 0.5$	0.5564	0.7363	0.4056	0.6266	0.0804
Lame $m = 2$ $\gamma = 4.0$	0.0997	0.0153	0.0078	n/a	0.3055
ZDT1	0.3924	0.0054	0.0020	n/a	0.2539
ZDT2	0.0465	0.0763	0.1669	n/a	0.9261

### 5.3. Computational Analysis of Preference-biased Approximations

Table 5.8.: Replacement strategies study – IGD. The table shows  $p$ -values of a post-hoc analysis of IGD. Each column lists the  $p$ -value of the comparison of BFP with respect to WIN and LED on the given problem. Significant results at a 5% level are highlighted in green for smallest and blue for second smallest  $p$ -value. All reported performance differences are significant with the exception of LED on DO2DK.

	WIN	LED
B1	0.0000	0.0000
B2	0.0001	0.0007
B3	0.0000	0.0000
B4	0.0000	0.0000
DO2DK $k = 2$ $s = 1$	0.0007	0.1168
DTLZ3	0.0013	0.0018

## 5. Preference-biased Pareto Front Approximations

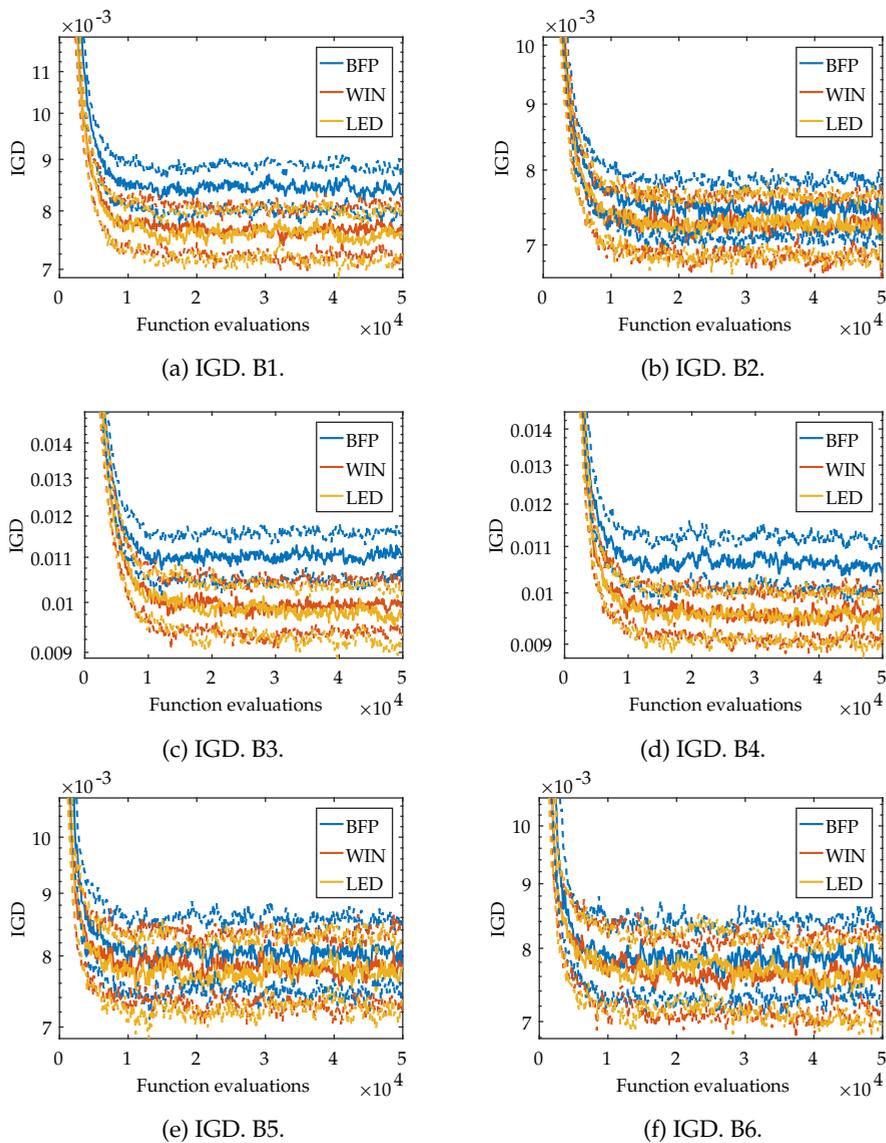


Figure 5.15.: Replacement strategies study – convergence of IGD for problems B1 to B6. Straight lines indicate median performance and dashed lines IQRs of performance.

### 5.3. Computational Analysis of Preference-biased Approximations

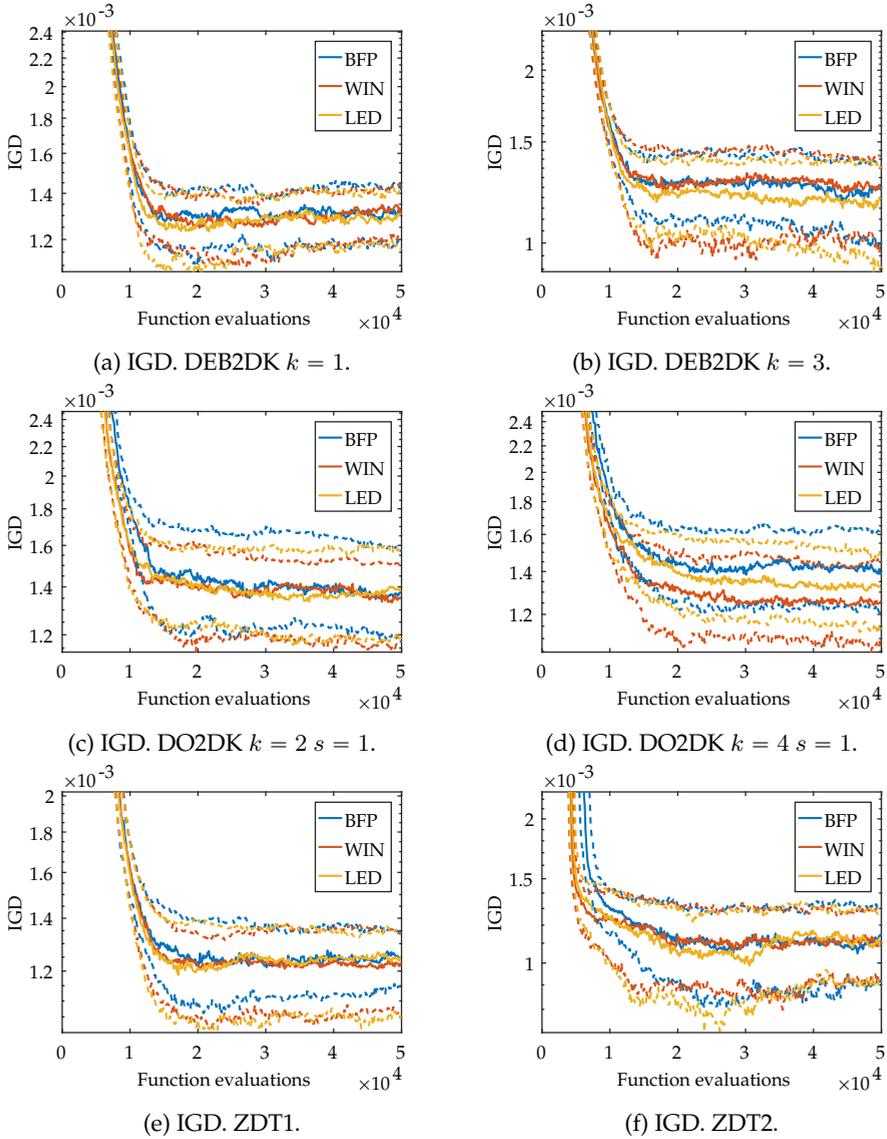


Figure 5.16.: Replacement strategies study – convergence of IGD for problems DEB2DK, DO2DK, ZDT1 and ZDT2. Straight lines indicate median performance and dashed lines IQRs of performance.

## 5. Preference-biased Pareto Front Approximations

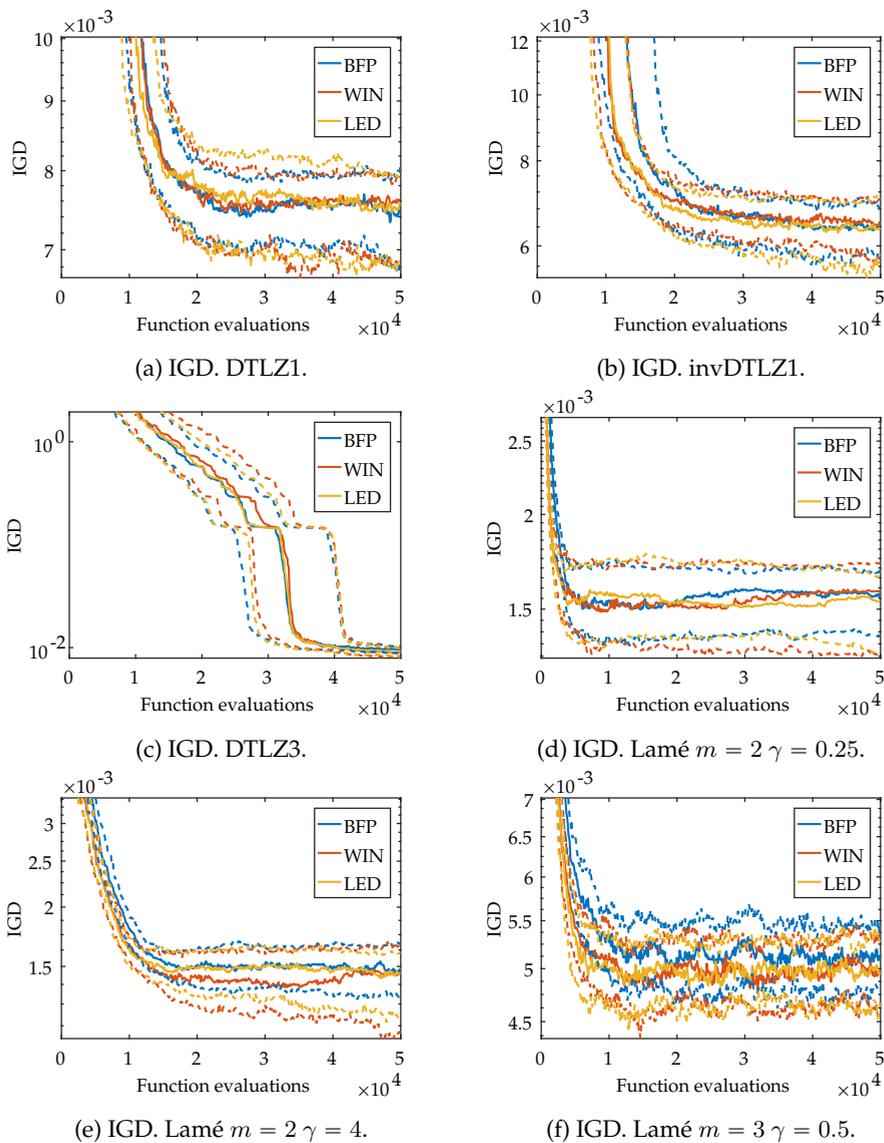


Figure 5.17.: Replacement strategies study – convergence of IGD for problems DTLZ1, invDTLZ1, DTLZ3 and the Lamé problems. Straight lines indicate median performance and dashed lines IQRs of performance.

### 5.3. Computational Analysis of Preference-biased Approximations

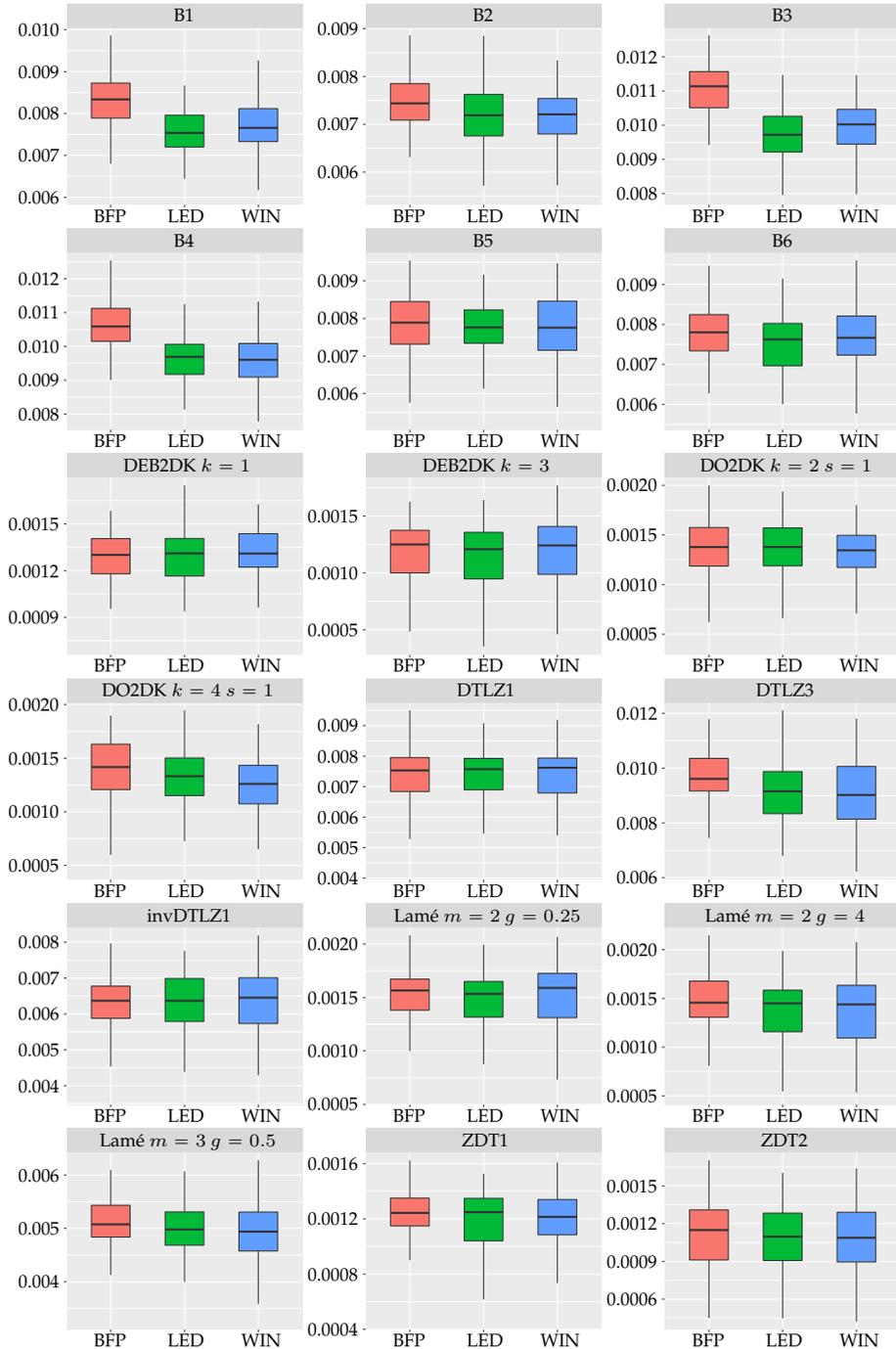


Figure 5.18.: Replacement strategies – boxplots of IGD after 50 000 function evaluations.

### 5.3.2. Quantitative Analysis of the No Preference Case

In case all Pareto optimal points are equally desirable, minimizing Equation (5.1) for a given  $N$  is expected to produce a uniform distribution of points on the Pareto front. The results of the study in the previous section confirm that ESPEA is well able to obtain such  $N$ -optimal distributions of points. In order to quantify how well ESPEA performs in comparison to existing finite set of points approaches, a quantitative study is conducted to benchmark ESPEA against established metaheuristics in MOO. This analysis also yields insights into how well the energy concept is suited to achieve uniform distributions of points in terms of established MOO performance indicators.

A population size of 100 was chosen for every algorithm used in the study, which is a common value used in the literature [DPAM02, ZLT01, ZK04, NDN<sup>+</sup>09]. Objective values were normalized (see Definition 52) for computing the Euclidean norm in Equation (5.1) based on minimum and maximum objective values of the current archive to eliminate the bias of objectives being measured on different scales (see Definition 51). Normalization is a common technique applied in MOO to achieve uniform finite set of points approximations [DPAM02, BNE07, DJ14]. Latin hypercube sampling [MBC79] was utilized in each algorithm to generate the initial population.

#### 5.3.2.1. Comparative Study Benchmark Algorithms

ESPEA is compared to seven other metaheuristics. The algorithms have been selected such that they represent a large array of different approaches and philosophies utilized in MOO to obtain finite set of points approximations. All chosen algorithms have been successfully applied to solve numerous academic and real-world optimization problems. They are established solving techniques that serve as reasonable benchmark to compare ESPEA to the state-of-the-art in MOO. The same search operators as presented in Section 5.3.1 are used with ESPEA. As suggested in the previous section, the WIN archive strategy is employed.

NSGA-II [DPAM02] is a generational EA that is one of the most frequently used metaheuristics in MOO and described in Algorithm 4. Nondominated sorting (Algorithm 2) and crowding distance (Algorithm 3) are two techniques that were introduced by NSGA-II to MOO and which have been utilized in many successive algorithms [TFD11, NDN<sup>+</sup>09, NLA<sup>+</sup>08]. Nondominated sorting (see Figure 2.4) divides the current population into subsets of points in the objective space called layers or fronts. Points that form part of a specific layer

### 5.3. Computational Analysis of Preference-biased Approximations

do not Pareto dominate each other. Each point of a given layer is Pareto dominated by at least one element of the previous layer save for the first layer, which consists of non-dominated elements.

---

**Algorithm 2:** nondominated sorting [DPAM02]

---

**Input :** Population  $P$

**Output:** Sorting  $(L_1, \dots, L_k)$

```

1  $k := 1$ 
2 while  $P \neq \emptyset$  do
3    $L_k := \{\mathbf{x} \in P \mid \nexists \mathbf{y} \in P : f(\mathbf{y}) <_p f(\mathbf{x})\}$ 
4    $P := P \setminus L_k$ 
5    $k := k + 1$ 
6 return  $(L_1, \dots, L_k)$ 

```

---

Crowding distance is a niching technique for estimating the density of the population in the objective space. The crowding distance of a point  $f(\mathbf{x})$  is the estimate of the volume of the largest cuboid that can be inscribed around  $f(\mathbf{x})$  such that no other population member is located in the cuboid (Figure 5.19). A large crowding distance implies that  $f(\mathbf{x})$  is located in a sparsely populated region of the objective space. Objective values are normalized based on the maximum and minimum objective values of the current population before crowding distance is computed. The crowding distance of extreme points is set to infinity such that they are retained for normalization in successive iterations.

---

**Algorithm 3:** crowding distance [DPAM02]

---

**Input :** Population  $P$

**Output:** Crowding distance  $\mathbf{c}$

```

1  $\mathbf{c} = \mathbf{0}^N$  // vector of crowding distances
2  $\mathbf{f}^{\max} := (\max_{\mathbf{x} \in P} f_1^{\mathbf{x}}, \dots, \max_{\mathbf{x} \in P} f_m^{\mathbf{x}})$  // max function values
3  $\mathbf{f}^{\min} := (\min_{\mathbf{x} \in P} f_1^{\mathbf{x}}, \dots, \min_{\mathbf{x} \in P} f_m^{\mathbf{x}})$  // min function values
4 forall  $i \in [m]$  do
5   Let  $s(j) := \arg \text{sort}((f_i^1, \dots, f_i^N), \leq, j)$ 
6    $c_{s(1)} := \infty$ 
7    $c_{s(N)} := \infty$ 
8   for  $j = 2$  to  $N - 1$  do
9      $c_{s(j)} := c_{s(j)} + \frac{f_i^{s(j-1)} - f_i^{s(j+1)}}{f_i^{\max} - f_i^{\min}}$ 
10 return  $\mathbf{c}$ 

```

---

NSGA-II performs nondominated sorting on the combined population of par-

## 5. Preference-biased Pareto Front Approximations

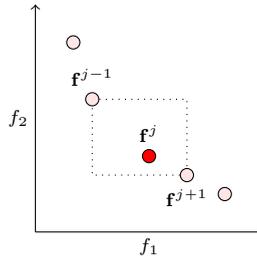


Figure 5.19.: Illustration of crowding distance. The points  $\mathbf{f}^{j-1}$  and  $\mathbf{f}^{j+1}$  mark the corners of the largest cuboid around  $\mathbf{f}^j$  that does not contain any other population member.

ents and offspring. The next generation's population is filled layer by layer until the current layer size exceeds the number of remaining open spots in the next generation. Next, the crowding distance of all elements of the current layer is computed. The elements that possess the largest crowding distance fill the remaining spots.

Parents are chosen in NSGA-II using binary tournament selection [Bri81]. Binary tournament selection randomly chooses two population members and compares them to each other using a predefined selection criterion. NSGA-II uses Pareto domination as first criterion. If the images of both individuals are non-dominated to each other crowding distance is compared as secondary criterion. Each tournament winner is chosen as parent and mated with another tournament winner. SBX and polynomial mutation were chosen as crossover and mutation operators. The same configuration as for ESPEA was used to provide a fair basis of comparison.

SPEA2 is a generational EA that uses an external archive, in which the best individuals that have been found during the search are stored. In each iteration, fitness values are assigned to the union of the current population and the archive. The fitness value is the sum of an individual's raw strength and its density value. The raw strength of an individual  $\mathbf{x}$  consists of the number of population and archive members by which  $f(\mathbf{x})$  is dominated. The density value is defined as 1 divided by the Euclidean distance in the objective space between  $f(\mathbf{x})$  and its  $k$ -nearest neighbor plus 2, whereas  $k$  is usually chosen as 1. The raw strength defines the rank of an individual similar to the number of the front in nondominated sorting. The density value serves as niching mechanism. A smaller fitness indicates that an individual is more desirable (see Algorithm 5).

Binary tournament selection using fitness values as selection criterion is ap-

### 5.3. Computational Analysis of Preference-biased Approximations

---

**Algorithm 4:** NSGA-II [DPAM02]
 

---

**Input :** MOOP  $f$   
**Output:** Pareto front approximation  $P$

- 1 Generate and evaluate initial population  $P$
- 2 **repeat**
- 3     Generate and evaluate offspring population  $Q$
- 4      $(L_1, \dots, L_k) := \text{nondominatedSorting}(P \cup Q)$
- 5      $P := \emptyset$
- 6      $i := 1$
- 7     **while**  $|P| + |L_i| \leq N$  **do**
- 8          $P := P \cup L_i$
- 9          $i := i + 1$
- 10     **if**  $|P| < N$  **then**
- 11          $\mathbf{c} := \text{crowdingDistance}(L_i)$
- 12          $\mathbf{s} := \text{argsort}(\mathbf{c}, \geq)$
- 13          $P := P \cup \{L_i(s_1), \dots, L_i(s_{|P|-N})\}$
- 14 **until** *stopping criterion*
- 15 **return**  $P$

---

plied for choosing parents in SPEA2. SBX and polynomial mutation were used for generating the offspring solution using the same configuration as in NSGA-II.

IBEA [ZK04] is a quality indicator-based generational EA (see Algorithm 6). A binary quality indicator in MOO is a function  $I(S, T)$  that maps to  $\mathbb{R}$ , where  $S$  and  $T$  are finite subsets of  $\mathbb{R}^m$ . The expression  $I(S, T)$  allows a quantitative statement about the relationship of  $S$  and  $T$ . Quality indicators are mainly used to compare finite set of points approximations. IGD (Definition 57) and RE (Definition 56) are examples for quality indicators. IBEA uses quality indicators to compute fitness values of population members. Indicator values between each pair of population members are computed. The fitness of an individual is then calculated by negating all its indicator values with respect to the other population members, dividing them by a scaling factor  $\kappa$ , further exponentiating the values and finally summing up all powers and negating the result (see Line 5 in Algorithm 6).

The value for the scaling constant was chosen as  $\kappa = 0.05$  as suggested in [ZK04]. Objective values are normalized based on the maximum and minimum objective values of the current population. Binary tournament selection with fitness values as selection criterion was applied for choosing parents. SBX and polynomial mutation were used for generating the offspring solution maintaining the same configuration as in NSGA-II. The hypervolume indicator as

## 5. Preference-biased Pareto Front Approximations

---

### Algorithm 5: SPEA2 [ZLT01]

---

**Input** : MOOP  $f$   
**Output**: Pareto front approximation  $A$

- 1 Generate and evaluate initial population  $P$
- 2  $A := \emptyset$
- 3 **repeat**
- 4      $Q := P \cup A$
- 5     Let  $F(\mathbf{x}) := |\{\mathbf{y} \in Q \mid f(\mathbf{y}) \prec_p f(\mathbf{x})\}| + \frac{1}{\min_{\mathbf{y} \in Q} \|f(\mathbf{x}) - f(\mathbf{y})\|_2 + 2}$
- 6     Let  $(\mathbf{x}^1, \dots, \mathbf{x}^{|Q|}) := Q$
- 7      $\mathbf{s} := \arg \text{sort}((F(\mathbf{x}^1), \dots, F(\mathbf{x}^{|Q|})), \leq)$
- 8      $A := \{Q(s_1), \dots, Q(s_N)\}$
- 9     Generate and evaluate offspring population and assign it to  $P$
- 10 **until** *stopping criterion*
- 11 **return**  $A$

---

presented in [ZK04] has been used in the study, which is defined as follows.

$$I_{HV}(S, T) = \begin{cases} HV(T, \mathbf{w}) - HV(S, \mathbf{w}) & \text{if } \forall \mathbf{v} \in T \exists \mathbf{u} \in S : \mathbf{u} \prec_p \mathbf{v} \\ HV(S + T, \mathbf{w}) - HV(S, \mathbf{w}) & \text{else,} \end{cases} \quad (5.17)$$

where  $HV(S, \mathbf{w})$  denotes the hypervolume of set  $S$  with respect to reference point  $\mathbf{w} \in \mathbb{R}^m$ .

SBX and polynomial mutation were used as evolutionary operators with IBEA. Both operators were configured in the same manner as for ESPEA. IBEA uses binary tournament selection with fitness values as selection criterion.

SMPSO [NDGN<sup>+</sup>09] is a particle swarm optimization (PSO) technique. Particle swarm algorithms (PSAs) are inspired by the natural phenomena of bird flocks and fish schools. In these swarm systems, the movement of each bird or fish is influenced by its relative position to its neighbors such that the swarm seemingly behaves as a single entity. PSAs mimic this behavior in guiding candidate solutions through the search space utilizing globally and locally best positions that have been attained by swarm members so far. Candidate solutions are referred to as particles in PSO and the swarm is equivalent to the population in EAs. In contrast to the population of an EAs, the swarm always maintains the same size throughout the entire execution of the algorithm. In this sense, no new particles are generated, existing candidate solutions are updated making their previous position lost when they move through the search space. Particles, however, memorize the best individual location they have attained in

### 5.3. Computational Analysis of Preference-biased Approximations

---

**Algorithm 6:** IBEA [ZK04]
 

---

**Input :** MOOP  $f$ , scaling factor  $\kappa$

**Output:** Pareto front approximation  $P$

```

1 Generate and evaluate initial population  $P$ 
2 repeat
3   Generate and evaluate offspring population  $Q$ 
4    $P := P \cup Q$ 
5   Let  $F(\mathbf{x}) := \sum_{\mathbf{y} \in P \setminus \{\mathbf{x}\}} -\exp\left(\frac{-I(\{f(\mathbf{y})\}, \{f(\mathbf{x})\})}{\kappa}\right)$  // assign
      fitness
6   while  $|P| > N$  do
7      $\mathbf{x}^* := \arg \min_{\mathbf{x} \in P} F(\mathbf{x})$ 
8      $P := P \setminus \{\mathbf{x}^*\}$ 
9     Let  $F(\mathbf{x}) := F(\mathbf{x}) + \exp\left(\frac{-I(\{\mathbf{x}^*\}, \{\mathbf{x}\})}{\kappa}\right)$  // update fitness
10 until stopping criterion
11 return  $P$ 

```

---

the search space and have access to global information about best positions achieved by other particles.

Let  $t$  be in the index of the current iteration. Each particle is characterized by its position  $\mathbf{x}^t$  in the decision space and its current velocity  $\mathbf{z}^t \in \mathbb{R}^n$ . The position of the particle is updated in each iteration by adding the velocity to its current position.

$$\mathbf{x}^t = \mathbf{x}^{t-1} + \mathbf{z}^t. \quad (5.18)$$

The velocity of the current iteration is computed through multiple calculations. First, the unconstrained velocity  $\mathbf{z}^{t,u}$  is obtained by weighing the velocity of the previous iteration by a factor  $w$  called inertia weight and adding two weighted directional vectors pointing to the particle's best individual or personal location  $\mathbf{x}^p$  and a globally best position  $\mathbf{x}^g$ . The unconstrained velocity is obtained by

$$\mathbf{z}^{t,u} = w\mathbf{z}^{t-1} + c_1 r_1 (\mathbf{x}^p - \mathbf{x}^{t-1}) + c_2 r_2 (\mathbf{x}^g - \mathbf{x}^{t-1}), \quad (5.19)$$

where  $r_1$  and  $r_2$  are uniform random numbers in  $[0, 1]$  and  $c_1$  and  $c_2$  are parameters that control the effect of the personal and global best. The unconstrained velocity is multiplied by a constriction coefficient  $\chi$  to obtain the constrained velocity

$$\mathbf{z}^t = \chi \cdot \mathbf{z}^{t,u} \quad (5.20)$$

with

$$\chi = \frac{2}{2 - \alpha - \sqrt{\alpha^2 - 4\alpha}} \quad (5.21)$$

## 5. Preference-biased Pareto Front Approximations

and

$$\alpha = \begin{cases} c_1 + c_2 & \text{if } c_1 + c_2 > 4 \\ 1 & \text{if } c_1 + c_2 \leq 4. \end{cases} \quad (5.22)$$

If the optimization problem is box constrained and the absolute value of a component of the velocity vector is larger than half of the length of the box in the current dimension, then the component's value is reset to half the length of the current dimension of the box keeping its original sign.

Algorithm 7 shows an outline of SMPSO.  $N$  denotes the swarm size. The algorithm features two archives  $A$  and  $B$ .  $B$  represents the personal memory. The element  $B(i)$  at position  $i$  of the archive is the best position of swarm member  $S(i) := \mathbf{x}^i$  that the particle has attained so far. The personal memory is updated, whenever  $\mathbf{f}^i$  Pareto dominates  $\mathbf{f}^{B(i)}$  in the current iteration. The crowding archive  $A$  contains the globally best points that have been obtained during the execution of SMPSO. The archive has a variable size and only contains nondominated points. Each updated particle is added to the archive if it is not dominated by any other archive member. If the archive exceeds its maximum size  $N$  after the insertion, crowding distances are computed and the archive member possessing the smallest crowding distance is eliminated. The call `updateSwarm` (Line 6) encompasses the steps explained in Equations (5.18) to (5.22) and is summarized in Algorithm 18. Each particle  $\mathbf{x}^i$  uses  $B(i)$  as personal best  $\mathbf{x}^p$  and a randomly chosen archive member as global best  $\mathbf{x}^g$ . After updating the swarm in Line 6, polynomial mutation is performed on each particle to broaden the search.

The algorithm was configured as proposed in the pre-existing implementation in the `jMetal` framework, which was developed by the designers of SMPSO. An inertia weight of  $w = 0.1$  was chosen. The parameters  $c_1$  and  $c_2$  were drawn as uniform random numbers in the interval  $[1.5, 2.5]$  for every velocity calculation. Polynomial mutation was configured in the same manner as for the other algorithms.

SMS-EMOA is a steady state EA that uses hypervolume contributions as niching technique (Algorithm 8). In each iteration, nondominated sorting is performed on the union of the population and the newly created candidate solution. For each member of the last front, the hypervolume that it contributes to the overall volume of the front is computed (Figure 4.2a). The candidate solution that possesses the smallest hypervolume contribution with respect to a reference point  $\mathbf{u}$  is deleted. The remaining individuals across all fronts are merged to form the population of the next generation. Objective values are normalized to mitigate the effect of different scalings during hypervolume contribution calculation. A reference point of  $\mathbf{w} = \mathbf{10}^m$  in the normalized objective space was chosen such that boundary points are always retained. A for-

### 5.3. Computational Analysis of Preference-biased Approximations

---

**Algorithm 7:** SMPSO [NDGN<sup>+</sup>09]
 

---

**Input :** MOOP  $f$ , inertia weight  $w$ , turbulence factors  $c_1, c_2$

**Output:** Pareto front approximation  $S$

```

1 Generate and evaluate initial swarm  $S := (\mathbf{x}^1, \dots, \mathbf{x}^N)$ 
2  $Z := (\mathbf{z}^1, \dots, \mathbf{z}^N)$  with  $\mathbf{z}^i = \mathbf{0}^n$  // velocities
3  $A := \{\mathbf{x}^i \in S \mid \nexists \mathbf{x}^j \in S : \mathbf{f}^j <_p \mathbf{f}^i\}$  // global best
4  $B := S$  // personal best
5 repeat
6    $(S, Z) := \text{updateSwarm}(S, Z, A, B, w, c_1, c_2)$ 
7   forall  $\mathbf{x}^i \in S$  do // update archive
8     if  $\forall \mathbf{y} \in A : \mathbf{f}(\mathbf{y}) \not<_p \mathbf{f}^i$  then
9        $A := \{\mathbf{y} \in A \mid \mathbf{f}^i <_p \mathbf{f}(\mathbf{y})\} \cup \{\mathbf{x}^i\}$ 
10      if  $|A| > N$  then
11         $\mathbf{c} = \text{crowdingDistance}(A)$ 
12         $A := A \setminus \{A(\arg \min_{i \in [N]} c_i)\}$  // remove worst
13      if  $\mathbf{f}^i <_p \mathbf{f}^{B(i)}$  then // update personal memory
14         $B(i) := \mathbf{x}^i$ 
15 until stopping criterion
16 return  $A$ 

```

---

mal description for the computation of hypervolume contributions using the Lebesgue measure  $\mathcal{L}$  is given in the appendix in Algorithm 19. SMS-EMOA used SBX and polynomial mutation with the same configuration as ESPEA. Binary tournament selection with hypervolume contributions as selection criterion was applied for the mating pool selection.

MOEA/D-DE or MOEAD in short [LZ09] is a decomposition-based EA (Algorithm 9). Decomposition-based algorithms divide the MOOP into scalar sub-problems for each of which a solution is computed. MOEAD uses the Chebyshev method (Definition 36) to decompose the MOOP into  $N$  SOOPs. Uniform weight vectors  $\{\lambda^1, \dots, \lambda^N\}$  taken from the unit simplex (cf. [DD98, Sec. 5]) are used with the Chebyshev decomposition to aim for generating a uniform approximation of the Pareto front. Each population member  $\mathbf{x}^i$  is associated with a neighborhood  $B^i$  that is composed of those population members whose weight vectors are closest to the vector  $\lambda^i$  of  $\mathbf{x}^i$  under the Euclidean norm.

In each iteration, a new candidates solution is generated for every population member  $\mathbf{x}^i$  using DE with the *rand/1/bin* reproduction scheme:

$$\mathbf{x}^m = \mathbf{x}^i + c_f(\mathbf{y}^1 - \mathbf{y}^2), \quad (5.23)$$

## 5. Preference-biased Pareto Front Approximations

---

### Algorithm 8: SMS-EMOA [BNE07]

---

**Input** : MOOP  $f$ , reference point  $\mathbf{w}$   
**Output**: Pareto front approximation  $P$

- 1 Generate and evaluate initial population  $P$
- 2 **repeat**
- 3     Generate and evaluate a single new candidate solution  $\mathbf{x}$
- 4      $P := P \cup \{\mathbf{x}\}$
- 5      $(L_1, \dots, L_k) := \text{nondominatedSorting}(P)$
- 6      $\mathbf{c} := \text{hypervolumeContributions}(L_k, \mathbf{w})$
- 7      $L_k := L_k \setminus \{L_k(\arg \min_{i \in [N]} c_i)\}$                                  // remove worst
- 8      $P := \{L_1, \dots, L_k\}$
- 9 **until** *stopping criterion*
- 10 **return**  $P$

---

where both  $\mathbf{y}^1$  and  $\mathbf{y}^2$  are elements of the mating pool. The mating pool is chosen with probability  $p_b$  as the neighborhood  $B^i$  of  $P(i)$  and with probability  $1 - p_b$  as the entire population  $P$ . The newly generated candidate solution  $\mathbf{y}$  is also mutated using polynomial mutation. The estimate of the ideal point  $\mathbf{u}^*$  required for computing the Chebyshev method is updated by replacing all of its components  $u_i^*$  by  $f_i^{\mathbf{y}}$  if  $f_i^{\mathbf{y}}$  is smaller than  $u_i^*$ . Next,  $\mathbf{y}$  replaces at most  $c_N$  population members. The candidate solution  $\mathbf{y}$  is compared to randomly drawn element  $\mathbf{z}$  of  $Q$ . If  $f(\mathbf{y})$  possesses a smaller scalarization value with respect to the Chebyshev method with  $\lambda^i$ , then  $\mathbf{y}$  replaces  $\mathbf{z}$  in the current population  $P$ . Afterwards,  $\mathbf{z}$  is removed from  $Q$ .

MOEAD was parametrized as suggested in [LZ09]. The neighborhood size was chosen to be 20, the selection probability was set to  $p_b = 0.9$  and the number of candidate solutions that are replaced at most to  $c_N = 2$ . The DE operator was parametrized with  $c_f = 0.5$ . Polynomial mutation was configured in the same manner as for the other algorithms.

The final algorithm that is considered for the study is NSGA-III [DJ14], an extension of NSGA-II that uses a decomposition-based niching mechanism instead of crowding distance. NSGA-III was specifically designed to obtain finite set of points approximations of MaOPs, however it has also shown to generate good results on MOOPs. The decomposition uses  $l$  reference points that are either provided by the DM or generated uniformly in the unit simplex by the same method that is applied in MOEAD. A single instance of the decomposed problem consists of finding a Pareto optimal objective vector  $\mathbf{u}$  that minimizes the perpendicular distance between  $\mathbf{u}$  and the line given by the reference point  $\lambda$  and the origin in the normalized objective space  $d^\perp(\mathbf{0}^m, \lambda, \mathbf{u})$ .

### 5.3. Computational Analysis of Preference-biased Approximations

---

#### Algorithm 9: MOEAD [LZ09]

---

**Input** : MOOP  $f$ , reference points  $\{\lambda\}_1^l$   
**Output**: Pareto front approximation  $P$

- 1 Generate initial population  $P := (\mathbf{x}^1, \dots, \mathbf{x}^N)$
- 2 Initialize  $B^1, \dots, B^N$
- 3  $\mathbf{u}^* := (\min_{\mathbf{x}^i \in P}(f_1^i), \dots, \min_{\mathbf{x}^i \in P}(f_m^i))$
- 4 **repeat**
- 5     **forall**  $\mathbf{x}^i \in P$  **do**
- 6         **if**  $\mathcal{U}(0, 1) < p_b$  **then**
- 7              $Q := B^i$
- 8         **else**
- 9              $Q := P$
- 10          $\mathbf{y} := \text{differentialEvolution}(\mathbf{x}^i, \text{random}(Q), \text{random}(Q))$
- 11          $\mathbf{y} := \text{mutate}(\mathbf{y})$
- 12          $\mathbf{u}^* := (\min(u_1^*, f_1^{\mathbf{y}}), \dots, \min(u_m^*, f_m^{\mathbf{y}}))$
- 13          $c := 0$
- 14         **while**  $(c < c_N) \wedge (Q \neq \emptyset)$  **do**
- 15              $\mathbf{z} := \text{random}(Q)$
- 16             **if**  $\Psi^{wc}(f(\mathbf{y})) < \Psi^{wc}(f(\mathbf{z}))$  *w.r.t.*  $\lambda^i$  **then**
- 17                  $P := (P \cup \{\mathbf{y}\}) \setminus \{\mathbf{z}\}$
- 18                  $c := c + 1$
- 19              $Q := Q \setminus \{\mathbf{q}\}$
- 20 **until** *stopping criterion*
- 21 **return**  $P$

---

In each iteration, nondominated sorting is applied to the union of population and offspring. The next generation population is filled front by front until the size of the current front  $L_i$  exceeds the number of remaining open spots in the next generation. If the next generation has not reached its maximum size the decomposition-based niching mechanism is applied to decide which members of  $L_i$  survive. The objective vectors of the elements in the next generation and the current front are normalized to prevent any effect of scaling on the distance calculation.

The next step associates each individual in the next generation and  $L_i$  with the reference point  $\lambda$  to which it has the smallest perpendicular distance in the objective space. The number of candidate solutions in the next generation that are associated with a given  $\lambda$  is referred to as niche count. The following selection mechanism is repeated until the next generation has reached the size  $N$ . A random index  $j$  is drawn among all reference points that possess the

## 5. Preference-biased Pareto Front Approximations

smallest niche count among all reference points. If no candidate solution in  $L_i$  is associated with  $\lambda^j$ , then  $\lambda^j$  is ignored until the next iteration. If the niche count of  $\lambda^j$  is zero, the element in  $L_i$  possessing the smallest perpendicular distance to  $\lambda^j$  is chosen. If the niche count of  $\lambda^j$  is greater than zero, an element in  $L_i$  that is associated with  $\lambda^j$  is randomly chosen. The chosen element is added to the next generation increasing the niche count of  $\lambda^j$  and removed from  $L_i$ .

Uniform reference points were generated as suggested in [DJ14] for two and three objectives. SBX and polynomial mutation were utilized to generate the offspring population. Polynomial mutation was configured in the same manner as for the other algorithms. A distribution index of 30 was chosen for SBX as suggested in [DJ14] to increase the likelihood of generating candidate solutions that are closer to the reference line. A crossover probability of 1.0 was used.

### 5.3. Computational Analysis of Preference-biased Approximations

---

**Algorithm 10: NSGA-III [DJ14]**


---

**Input :** MOOP  $f$ , reference points  $\{\lambda\}_1^l$

**Output:** Pareto front approximation  $P$

```

1 Generate initial population  $P$ 
2 repeat
3   Generate offspring population  $Q$ 
4    $(L_1, \dots, L_k) := \text{nondominatedSorting}(P \cup Q)$ 
5    $P := \emptyset$ 
6    $i := 1$ 
7   while  $|P| + |L_i| \leq N$  do
8      $P := P \cup L_i$ 
9      $i := i + 1$ 
10  if  $|P| < N$  then
11     $\text{normalize}(P \cup L_i)$ 
12     $C^P(\lambda) := \{\mathbf{x} \in P \mid \lambda = \arg \min_{\lambda^j \in \{\lambda\}_{j=1}^l} d^\perp(\mathbf{0}^m, \lambda^j, f(\mathbf{x}))\}$ 
13     $C^{L_i}(\lambda) := \{\mathbf{x} \in L_i \mid \lambda = \arg \min_{\lambda^j \in \{\lambda\}_{j=1}^l} d^\perp(\mathbf{0}^m, \lambda^j, f(\mathbf{x}))\}$ 
14     $I := [l]$ 
15    while  $|P| < N$  do
16       $j := \text{random}(\{a \in I \mid |C^P(\lambda^a)| = \min_{b \in I} (|C^P(\lambda^b)|)\})$ 
17      if  $C^{L_i}(\lambda^j) = \emptyset$  then
18         $I := I \setminus \{j\}$ 
19      else
20        if  $|C^P(\lambda^j)| = 0$  then
21           $\mathbf{y} := \min_{\mathbf{x} \in C^{L_i}(\lambda^j)} d^\perp(\mathbf{0}^m, \lambda^j, f(\mathbf{x}))$ 
22          else
23             $\mathbf{y} := \text{random}(C^{L_i}(\lambda^j))$ 
24           $P := P \cup \{\mathbf{y}\}$ 
25           $C^P(\lambda^j) := C^P(\lambda^j) \cup \{\mathbf{y}\}$ 
26           $C^{L_i}(\lambda^j) := C^{L_i}(\lambda^j) \setminus \{\mathbf{y}\}$ 
27 until stopping criterion
28 return  $P$ 

```

---

## 5. Preference-biased Pareto Front Approximations

### 5.3.2.2. Comparative Study Test Problems

The computational study is split into two parts. The first experiment focuses on the general performance of each algorithm on a heterogeneous test bed of problems. Each problem was selected such that Pareto fronts of various types, i.e. convex, concave, disconnected etc., are present in the study. In total, 24 problems are considered of which all 18 from the study of the replacement strategies are retained (Table 5.2). Thereby, six new problems are added to the simulation (see Table 5.9). DEB3DK [BDDO04] is a three objective extension of DEB2DK. The problem also possesses a parameter  $k$  for controlling the number of convex bulges of the Pareto front. The total number of bulges is given by  $2^k$ . The parameter  $k$  induces a strong deformation of the Pareto front, which leads to the surface that forms the front becoming partially dominated. These dominated parts can be identified as holes in Figure 5.20. Niching techniques must avoid these holes in order to generate a genuine approximation of the Pareto front.

DTLZ5 is a three objective problem, whose Pareto front is an arc of a circle [DTLZ05]. It is therefore considered degenerate, since the front is a manifold of dimension 1 (see Figure 5.21). The objective space *above* the front forms a wedge, effectively squeezing the population as it approaches the front. Redundant objectives as in the case of DTLZ5 might deter niching techniques from finding a uniform finite set of points approximation. The Pareto front of DTLZ7 is composed of four disconnected patches [DTLZ05]. As in the case of DEB3DK, the feasible region of the objective space between the patches is Pareto dominated (see Figure 5.20).

ZDT3's Pareto front is a sinusoidal wave that expands towards the extreme point of the first objective. The front is therefore disconnected, since the segments in which the wave exhibits an upward movement are dominated. The Pareto front of ZDT6 possesses a concave shape similar the one of ZDT2 (Figure 5.2). The ZDT6 problem, however, is more difficult to solve, since small steps in the decision space result in large jumps in the objective space. Search operators might therefore be deterred from successfully generating candidate solutions in sparsely crowded regions.

The second experiment focuses on the influence of the shape of the Pareto front on the algorithms' performance. In order to isolate the effect of Pareto front curvature on the performance, random instances of the Lamé with  $m = 2$  and DEB2DK problems are considered. Recall that the parameter  $\gamma$  of the Lamé problem changes the curvature of the front. Values smaller than 1 imply a convex curvature and values greater than one a concave curvature (see Figure 5.5). 100 uniform random numbers were drawn in the interval (0.25,1) and (1,4), respectively, to obtain 100 convex and 100 concave Lamé instances. Recall that the parameter  $k$  sets the number of knees of the Pareto front of DEB2DK.

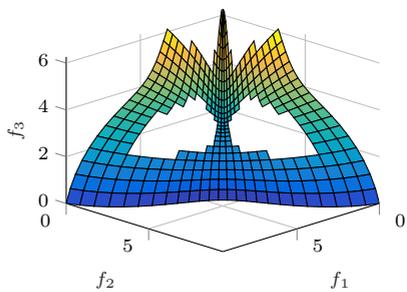
### 5.3. Computational Analysis of Preference-biased Approximations

Table 5.9.: Additional problems utilized in the ESPEA no preference study. The table features the problem names including their parametrization, the source of publication, the number of objectives  $m$  and the number of decision variables  $n$ . For the remaining problems, see Table 5.2.

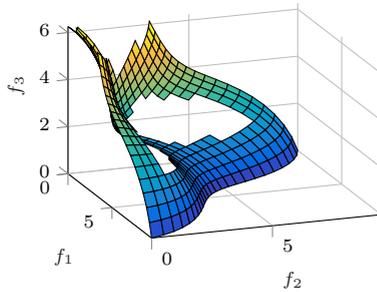
Name	Source	$m$	$n$
DEB3DK $k = 1$	[BDDO04]	3	30
DEB3DK $k = 2$	[BDDO04]	3	30
DTLZ5	[DTLZ05]	3	12
DTLZ7	[DTLZ05]	3	22
ZDT3	[ZDT00]	2	30
ZDT6	[ZDT00]	2	10

For DEB2DK, 100 problem instances were generated by selecting each value  $k \in \{1, \dots, 10\}$  ten times. The proposed test bed consists of three test instances: convex, concave and mixed curvature.

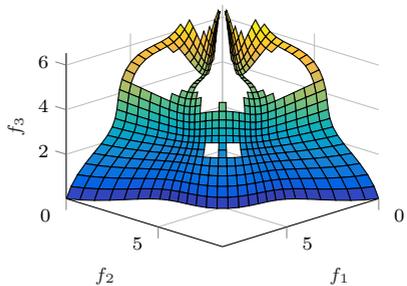
5. Preference-biased Pareto Front Approximations



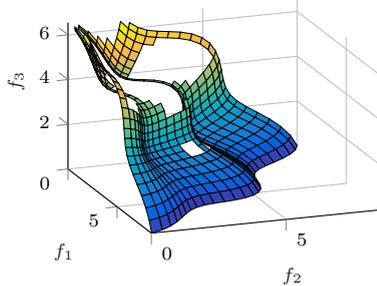
(a) DEB3DK  $k = 1$  (front).



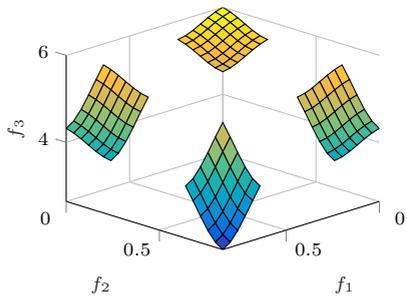
(b) DEB3DK  $k = 1$  (side).



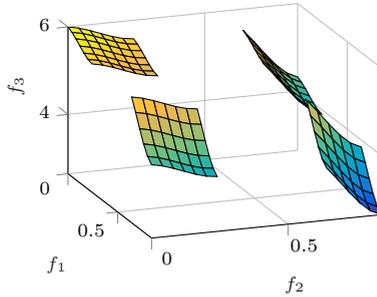
(c) DEB3DK  $k = 2$  (front).



(d) DEB3DK  $k = 2$  (side).



(e) DTLZ7 (front).



(f) DTLZ7 (side).

Figure 5.20.: Pareto fronts of the problems DEB3DK and DTLZ7 from the front and the side.

### 5.3. Computational Analysis of Preference-biased Approximations

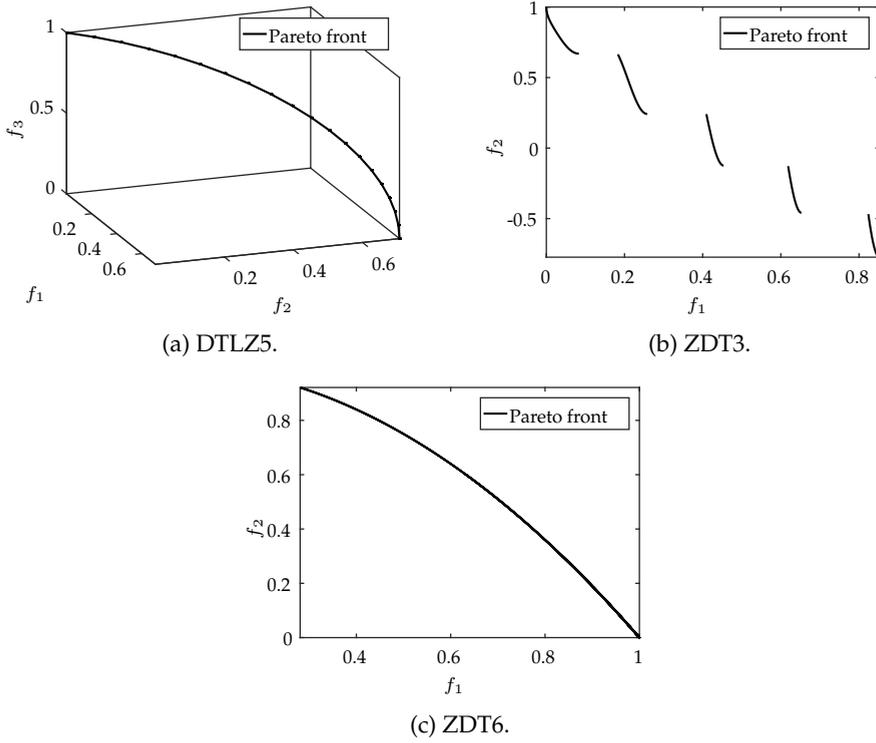


Figure 5.21.: Pareto fronts of the problems DTLZ5, ZDT3 and ZDT6.

## 5. Preference-biased Pareto Front Approximations

### 5.3.2.3. Comparative Study Performance Indicators

IGD is chosen as sole performance indicator in the first experiment (see Definition 57). The IGD metric is one of the most common performance indicators to benchmark finite set of points algorithms, since it captures both convergence and diversity [BSS15, LZ09, DJ14]. Convergence measures how close an approximation is to the Pareto front and diversity how well-spread its points are on the front. The computation of IGD requires a finite reference set (see Equation (5.16)). Therefore, a uniform discretization of the Pareto front of each test problem was generated. Each discretization possesses between 10 000 and 20 000 points such that the entire Pareto front is sufficiently covered.

In the second experiment, IGD and the spread metric (Definition 58) are considered. The spread metric is a performance indicator for two objective problems that measures the average distance between consecutive elements of the Pareto front approximation. Recall that the lexicographic order allows a total ordering of a two objective Pareto front either from left to right or from right to left. The spread metric consists of the average distance of successive objective vectors when counted from left to right.<sup>5</sup> A smaller spread signalizes a higher quality of approximation. Objective values are normalized before distances are computed.

**Definition 58** (spread [DPAM02]). *Let  $S \subset \mathbb{R}^m$  and let  $S$  be ordered lexicographically with  $S := \{\mathbf{u}^1, \dots, \mathbf{u}^{|S|}\}$ . The spread  $SPREAD(S)$  is defined as*

$$SPREAD(S) = \frac{1}{|S|} \sum_{i=1}^{|S|-1} \|\mathbf{u}^i - \mathbf{u}^{i+1}\|_2. \quad (5.24)$$

The spread metric does not require a reference set such as the IGD or RE. It is only a meaningful indicator for comparing algorithm performances if the Pareto front approximations cover the entire Pareto front. Additionally, similarly to the RE indicator, it is only viable to compare sets of equal sizes. For example, a Pareto front approximation that consists of a single point has a spread of zero, irrespective of its location in the objective space. For these reasons, a convergence analysis of the spread metric is omitted. The spread metric measures only the diversity of the Pareto front approximation and not its convergence to the true front.

---

<sup>5</sup>Equivalently, successive objective vectors may also be counted from right to left.

#### 5.3.2.4. Discussion of the Comparative Study Results

All algorithms were run 100 times on each test problems for 50 000 function evaluations, which are equivalent to 500 generations.<sup>6</sup> The results of the first experiment are analyzed first. Performance across the 500 generations with respect to the IGD indicator is depicted in Figures 5.22 to 5.29. The figures depicting the performance on each problem have been split into two plots, respectively, so the differences between the algorithms are better visible. ESPEA is contained in each of these plots for an easier comparison. The figures show magnified picture details for a better analysis of the convergence behavior.

The figures show vast performance differences between the algorithms on the test problems. ESPEA is among the top performing algorithms on the B problem family. SPEA2, however, is the algorithm that performs best and the only one that beats ESPEA on all problem instances. ESPEA is the second-best performing algorithm on B1, B2 and B3. In addition to SPEA2, ESPEA is outperformed by IBEA and NSGA-III on B4. On top of that, NSGA-III's population achieves smaller IGDs on B5 and B6. The IQRs of the IGDs, however, border each other on B5 and B6 suggesting that the actual performance gap between both algorithms is narrow.

The performance of NSGA-II exhibits a pronounced drop on B2 and B4 after decreasing sharply in the beginning. Such performance drops may be explained by the niching mechanism being unable to evenly distribute points across the front. In early iterations, the population approaches the Pareto front uniformly. Once the population is close to the front, the niching mechanism is unable to handle the geometry of the front, which leads to an unequal distributions of points across the front.

ESPEA also obtains good results on the DEB2DK problems. It shares the second-best performance with NSGA-III and is only clearly outperformed by SMPSO. In comparison, ESPEA achieves better results than SMPSO on all instances of the B problem family. On the DEB3DK problem instances, ESPEA is the fourth ( $k = 1$ ) and third best algorithm ( $k = 2$ ) and outperformed by NSGA-III, SMS-EMOA (only for  $k = 2$ ) and SPEA2, however in early iterations, ESPEA converges faster than both NSGA-III and SPEA2. ESPEA obtains the second smallest IGD on the DO2DK instances and is only outperformed by SMPSO.

Results on the DTLZ problem family are mostly positive. While ESPEA is outperformed by NSGA-III, SMS-EMOA and SPEA2 on DTLZ1, it achieves the second-best results on the inverted DTLZ1 being only bested by SPEA2. ESPEA belongs to the group of best performing algorithms on DTLZ3 and shares

---

<sup>6</sup>In generational EAs, performance is only measured after each iteration, in which function evaluations equivalent to the population size are performed.

## 5. Preference-biased Pareto Front Approximations

the second place with SPEA2 on DTLZ5. On DTLZ7, however, ESPEA is outperformed by the majority of the other algorithms. The IQR of IGD attained by ESPEA, however contains the IQRs of all other algorithms besides SPEA2. This suggests that although the performance gap is large, it may not be significant.

On the Lamé problem instances, ESPEA is the overall top performing algorithm. For  $m = 2$  and  $\gamma = 0.25$  it shares the first place with SMPSO. SMPSO outperforms ESPEA for  $m = 2$  and  $\gamma = 4$ , however ESPEA obtains the best results for  $m = 3$  and  $\gamma = 0.25$ , while SMPSO is only third best. ESPEA exhibits a similarly strong performance on the ZDT problems. It belongs to the algorithms that achieve the smallest IGD on ZDT2, ZDT3 and ZDT6 and is only outperformed on ZDT1.

The convergence plots reveal that ESPEA is competitive when compared to the state-of-the-art in MOO. On the vast majority of the considered benchmark problems, ESPEA ranks among the top three performing algorithms. The narrow bounds of the IQRs of the IGDs obtained by ESPEA also suggests that the algorithm delivers stable results. The performance of SMPSO, which often outperforms ESPEA, for example, appears more unstable, since its IQRs are larger on average. In addition, the performance gaps to algorithms that achieve better results than ESPEA is often narrow. The median IGD of ESPEA's population constantly decreases across the 50 000 function evaluations on all problem instances, which further underlines the stability of ESPEA's results. Regarding the problem characteristics, it appears though that ESPEA performs moderately worse on three dimensional problems compared to two dimensional problems. One reason for this observed behavior could be that the size of the boundary of the Pareto front increases as the number of objectives increases. Points that lie on the boundary possess no neighboring points beyond the boundary, leading them to induce less energy into the archive than interior points. This in turn may result in ESPEA putting a stronger focus on approximating the boundary compared to the interior of the front.

Table 5.10 shows the IGD after 50 000 function evaluations for all eight algorithms. The figures confirm the analysis of the convergence plots. ESPEA, SMPSO and SPEA2 are the top performing algorithms after all function evaluations have been exhausted. NSGA-III also achieves good results, but appears to perform weaker than the top three algorithms. SMS-EMOA scores the smallest IGD on two test problems and the second smallest IGD on one problem. IBEA attains a second best performance on B4. MOEAD and NSGA-II achieve no best or second best place in the performance ranking and can therefore be considered outmatched.

The performance differences observed in the convergence behavior and after the completion of the entire 50 000 function evaluations raise the question, whether these differences may be traced to the fundamental properties of the

### 5.3. Computational Analysis of Preference-biased Approximations

problems and if these are linked to the algorithms niching techniques. ESPEA and SPEA2 appear to achieve good results irrespective of the underlying problem. This may be attributed to their niching mechanisms, which are both not largely influenced by the geometry of the front. The niching mechanisms of ESPEA and SPEA2 both depend on the inverse Euclidean distance between population members. For a given distance  $d$  and Pareto optimal point  $\mathbf{u}$  another Pareto optimal point  $\mathbf{v}$  can be found, such that the distance between  $\mathbf{u}$  and  $\mathbf{v}$  equals  $d$  irrespective of the geometry of the front as long as it is connected. In contrast, hypervolume-based niching techniques are largely affected by the front geometry [IISN17]. It is a well-known fact, that hypervolume understates the importance of points close to the boundary on concave Pareto fronts, since they contribute less to the overall hypervolume of the population (see Figure 4.2). This is clearly reflected by the data, which shows that SMS-EMOA performs worse compared to other algorithms on problems with concave Pareto fronts.

ESPEA, however, tends to overstate the importance of points that lie on the boundary of the Pareto front, since such points lie farthest away from other population members. This leads to more points of the Pareto front approximation being located on the boundary of the Pareto front. Said fact may explain, why SPEA2 tends to outperform ESPEA on nearly all three objective problems. In contrast, ESPEA achieves better results than SPEA2 on most of the two objective problems. ESPEA's niching mechanism, which takes the distances between all population members into account – and not just the nearest neighbors – is able to distribute the points more evenly on the front.

Interestingly, ESPEA achieves the worst results on DTLZ7 being outperformed by all algorithms besides IBEA. DTLZ7 features a disconnected Pareto front. A qualitative analysis of the Pareto front approximations obtained by ESPEA has revealed that the algorithm sometimes only approximates either of the four patches that form the front (Figure 5.20e). This may be attributed to the strong selection pressure induced by ESPEA's steady state approach. In case the algorithm converges to early to only either of the patches, it might be incapable of branching out the approximation to the other patches.

SMPSO performs notably better on two objective compared to three objective problems. Since global leaders are selected at random from the crowding archive, the algorithm might struggle to achieve a stable distribution of points on the front. Instead, particles are likely to reposition themselves continuously leading to a non-uniform distribution. Crowding distance is also known to be an insufficient density estimator in three and higher dimensions [KD06]. This is also reflected by the performance of NSGA-II, which performs better on three objectives than on two objective problems.

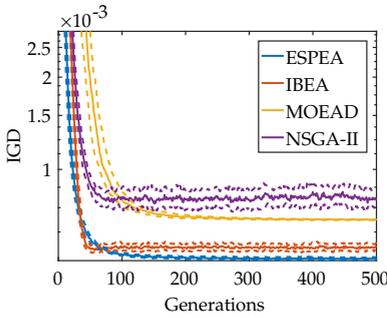
## 5. Preference-biased Pareto Front Approximations

Decomposition-based MOEAs have shown to be more efficient in obtaining finite set of points approximations of MaOPs compared to traditional domination-based techniques [ITN08, DJ14]. The reason for this is that niching techniques that work well for two or three objectives fail to estimate the population density in higher dimensions as the ratio of nondominated to dominated/dominating candidate solutions increases. This may serve as an explanation to why NSGA-III appears to perform better on the three compared to the two objective problems.

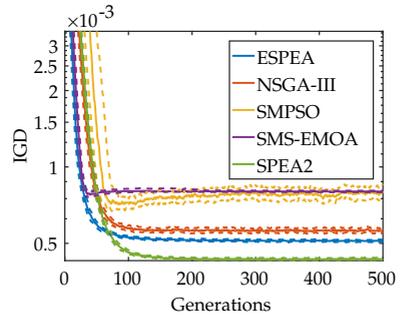
In order to confirm whether the observed performance differences are significant, a statistical analysis is performed. The same procedure that has been presented in the previous section is applied. Table G.15 reveals that the hypothesis of normally distributed data is rejected for at least one algorithm on all problem instances. A subsequent Kruskal-Wallis reveals that significant performance differences exist on all problem instances. Results of the post-hoc analysis are only reported for the comparison of ESPEA to the other seven algorithms, since the focus of this study is evaluating ESPEA against state-of-the-art finite set of points algorithms.

The results in Table 5.11 confirm that ESPEA outperforms the majority of the other algorithms on nearly all test problems. ESPEA only obtains once results that are significantly worse than those of IBEA (B4), MOEAD and NSGA-II (both DTLZ7). On all other problem instances, ESPEA surpasses the three algorithms with confidence. ESPEA also significantly outperforms SMS-EMOA on the majority of the 24 test problems. NSGA-III and SMPSO surpass ESPEA on multiple problems, still ESPEA achieves on the majority of the test instances better results. The performance of SPEA2 and ESPEA appears to be on par.

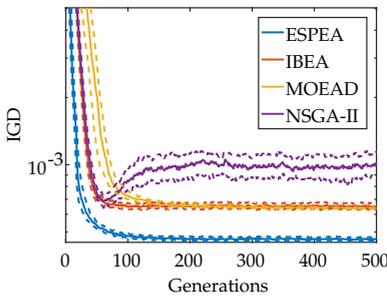
### 5.3. Computational Analysis of Preference-biased Approximations



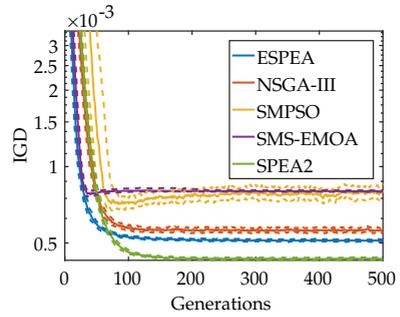
(a) IGD. B1.



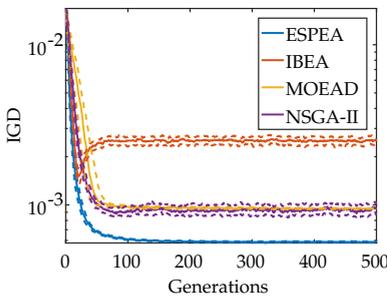
(b) IGD. B1.



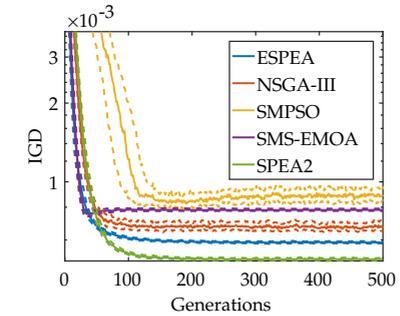
(c) IGD. B2.



(d) IGD. B2.



(e) IGD. B3.



(f) IGD. B3.

Figure 5.22.: Comparative study of ESPEA – convergence of IGD for problems B1 to B3. Straight lines indicate median performance and dashed lines IQRs of performance.

## 5. Preference-biased Pareto Front Approximations

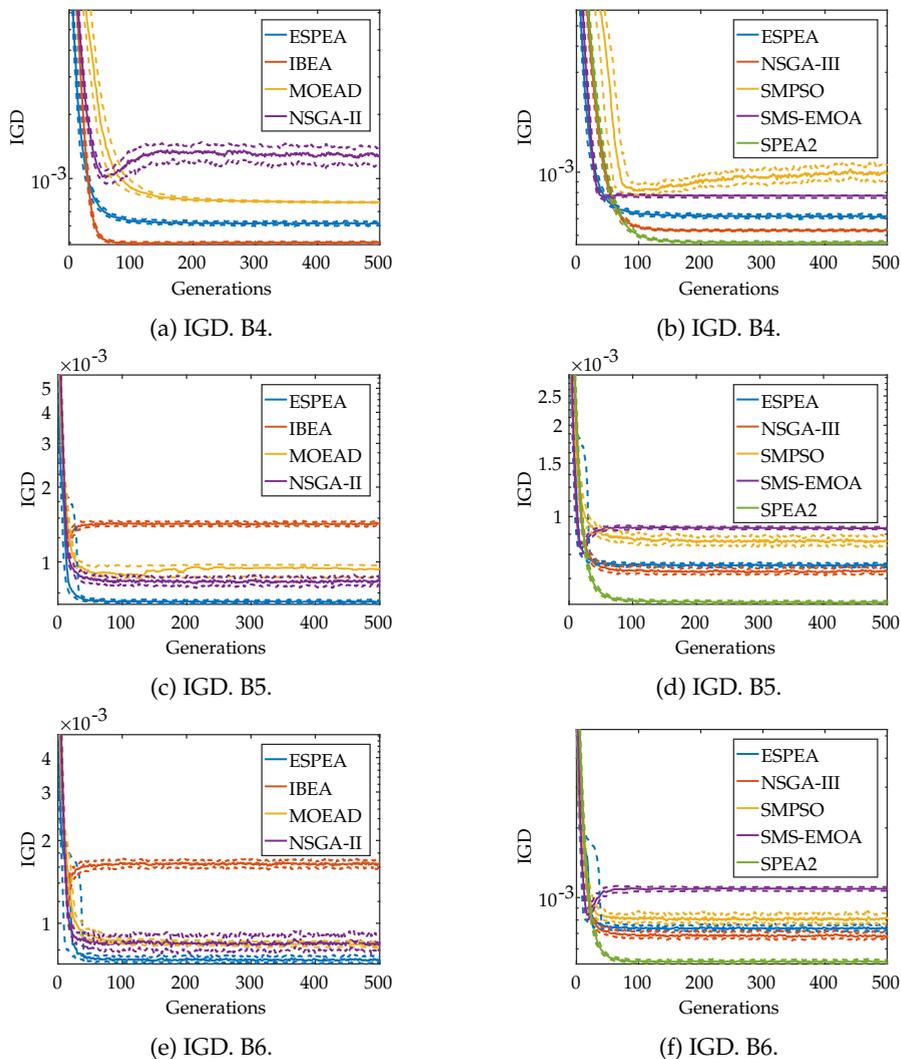


Figure 5.23.: Comparative study of ESPEA – convergence of IGD for problems B4 to B6. Straight lines indicate median performance and dashed lines IQRs of performance.

### 5.3. Computational Analysis of Preference-biased Approximations

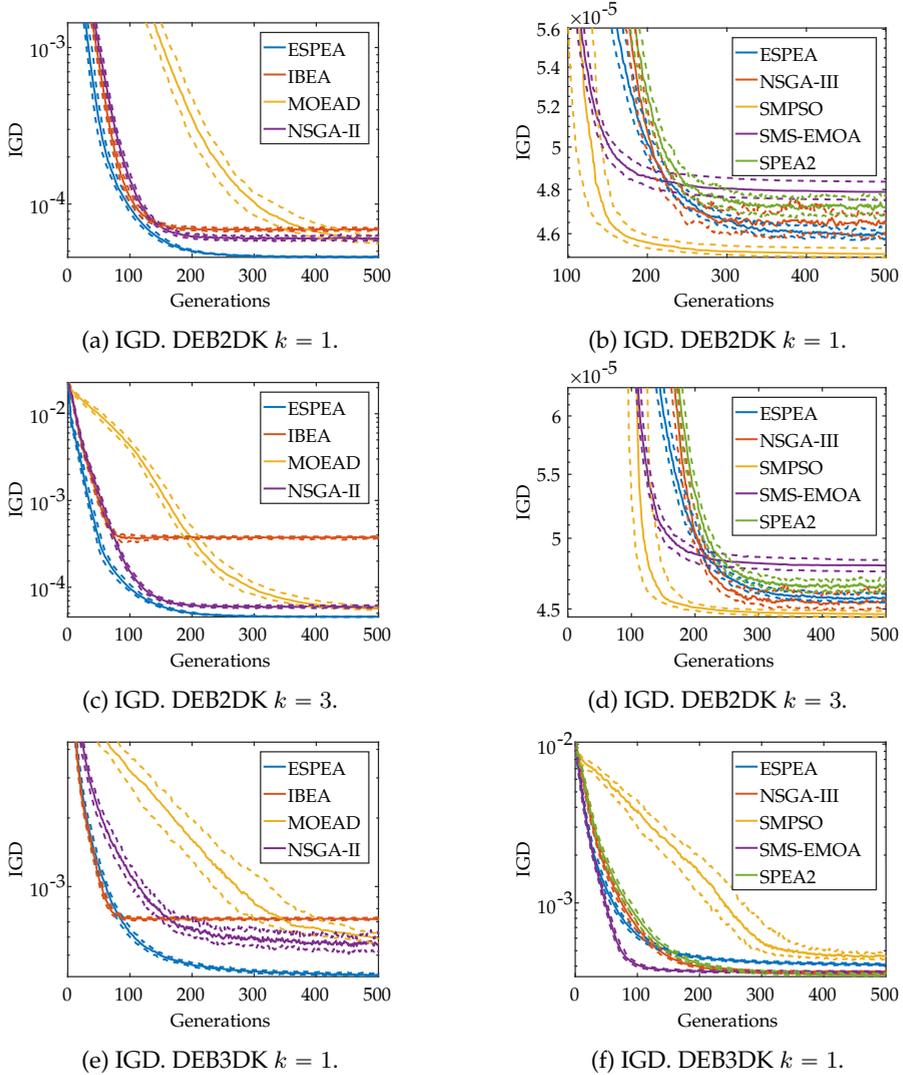
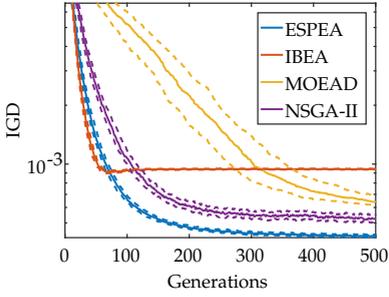
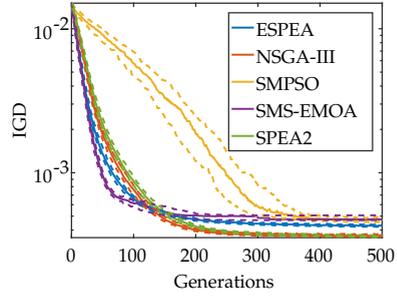


Figure 5.24.: Comparative study of ESPEA – convergence of IGD for problems DEB2DK and DEB3DK. Straight lines indicate median performance and dashed lines IQRs of performance.

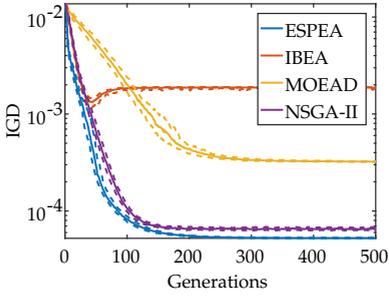
5. Preference-biased Pareto Front Approximations



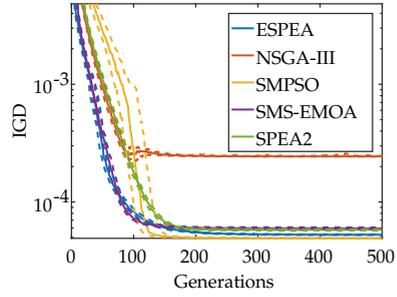
(a) IGD. DEB3DK  $k = 2$ .



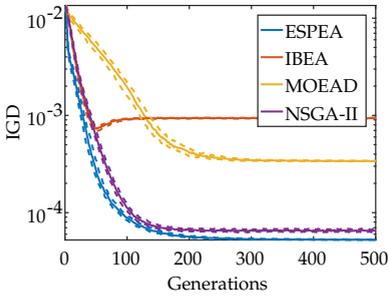
(b) IGD. DEB3DK  $k = 2$ .



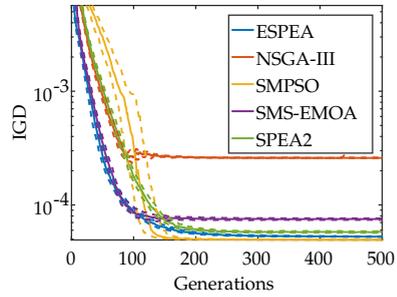
(c) IGD. DO2DK  $k = 2$   $s = 1$ .



(d) IGD. DO2DK  $k = 2$   $s = 1$ .



(e) IGD. DO2DK  $k = 4$   $s = 1$ .



(f) IGD. DO2DK  $k = 4$   $s = 1$ .

Figure 5.25.: Comparative study of ESPEA – convergence of IGD for problems DEB3DK and DO2DK. Straight lines indicate median performance and dashed lines IQRs of performance.

### 5.3. Computational Analysis of Preference-biased Approximations

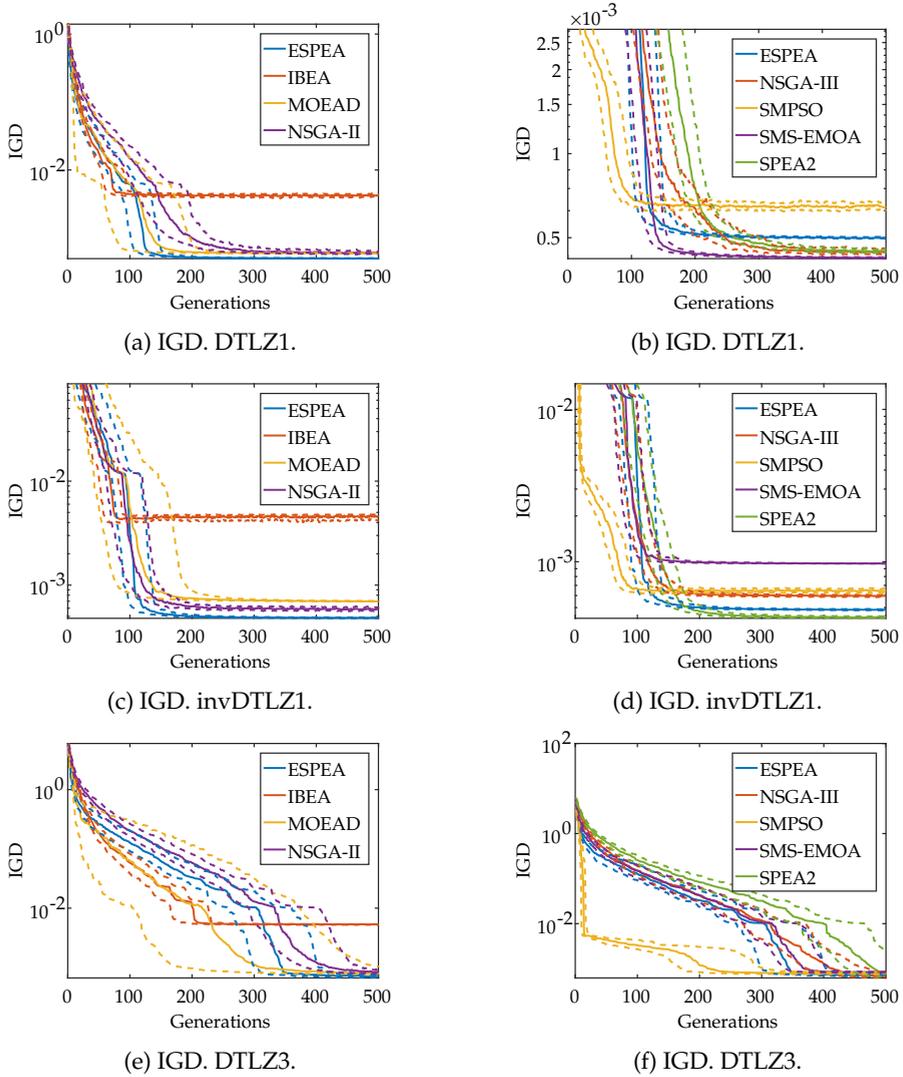


Figure 5.26.: Comparative study of ESPEA – convergence of IGD for problems DTLZ1, invDTLZ1 and DTLZ3. Straight lines indicate median performance and dashed lines IQRs of performance.

## 5. Preference-biased Pareto Front Approximations

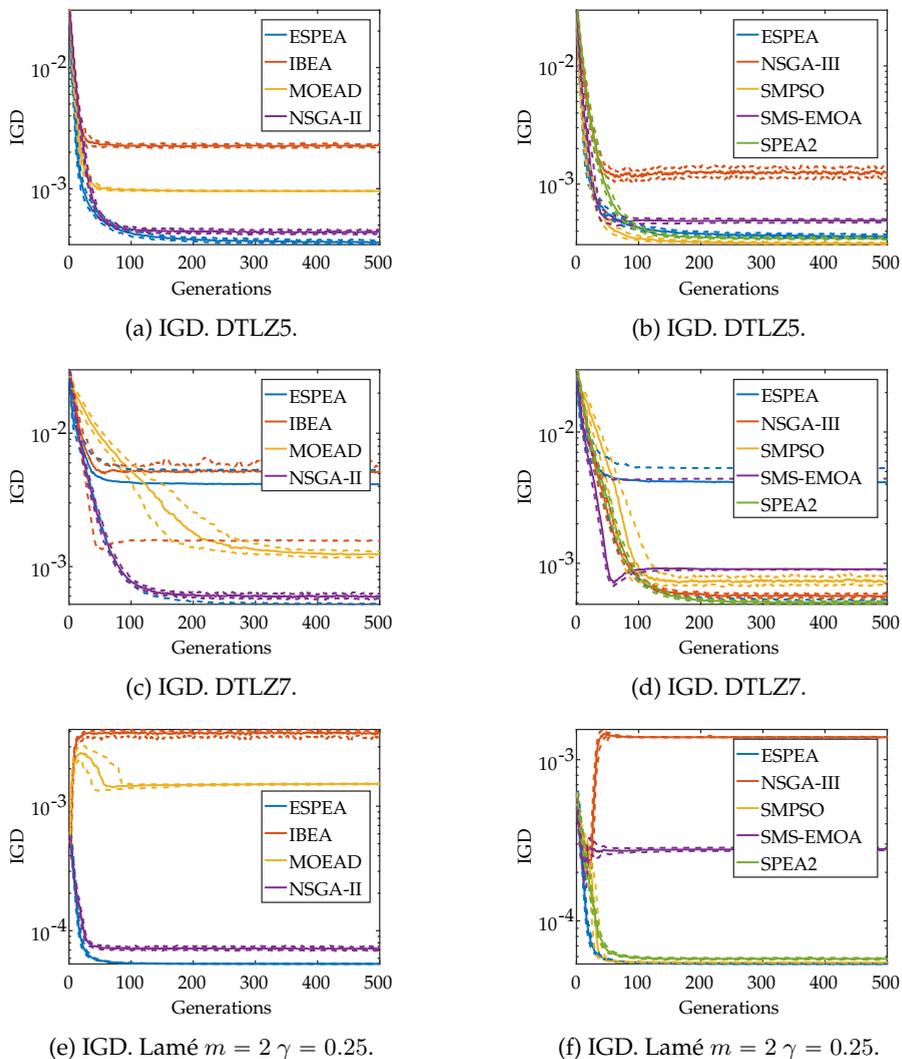
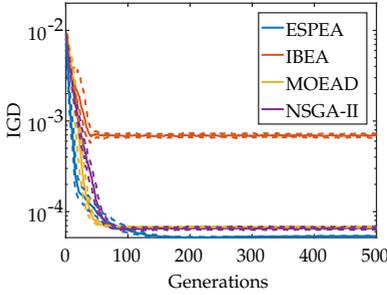
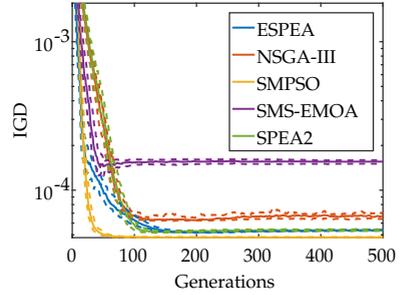


Figure 5.27.: Comparative study of ESPEA – convergence of IGD for problems DTLZ5, DTLZ7 and a two objective Lamé problem. Straight lines indicate median performance and dashed lines IQRs of performance.

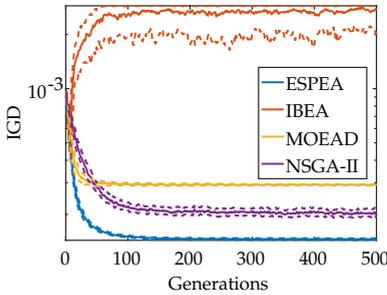
### 5.3. Computational Analysis of Preference-biased Approximations



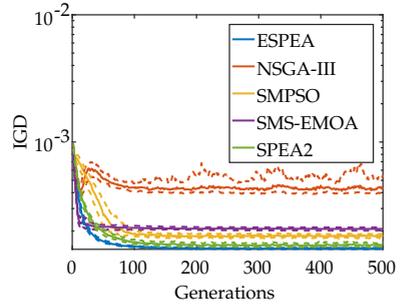
(a) IGD. Lamé  $m = 2 \gamma = 4$ .



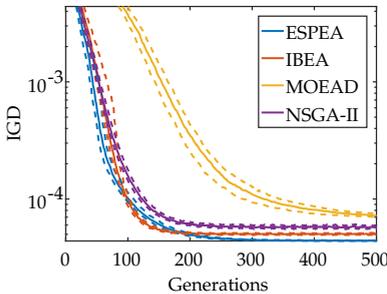
(b) IGD. Lamé  $m = 2 \gamma = 4$ .



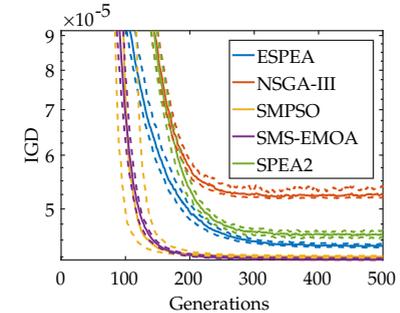
(c) IGD. Lamé  $m = 3 \gamma = 0.5$ .



(d) IGD. Lamé  $m = 3 \gamma = 0.5$ .



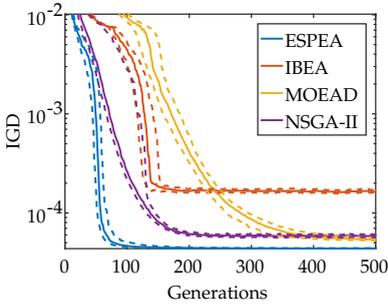
(e) IGD. ZDT1.



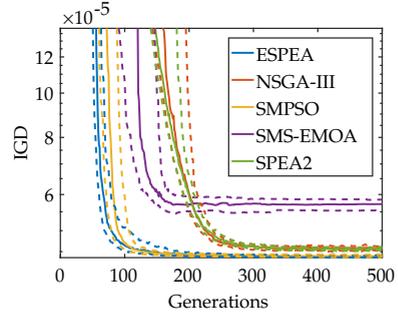
(f) IGD. ZDT1.

Figure 5.28.: Comparative study of ESPEA – convergence of IGD for Lamé problems and problem ZDT1. Straight lines indicate median performance and dashed lines IQRs of performance.

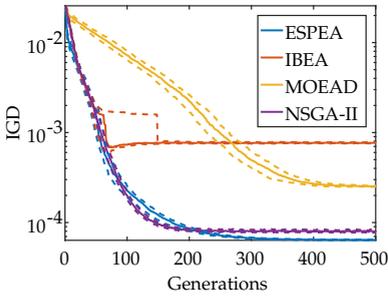
5. Preference-biased Pareto Front Approximations



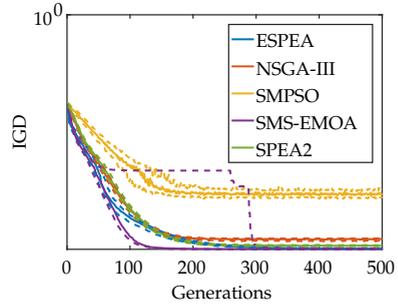
(a) IGD. ZDT2.



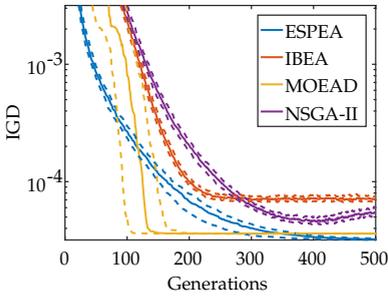
(b) IGD. ZDT2.



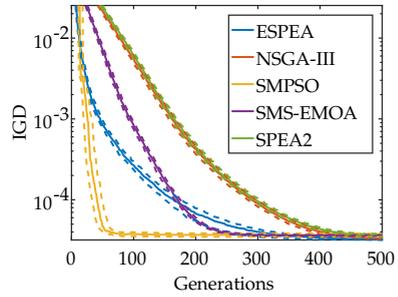
(c) IGD. ZDT3.



(d) IGD. ZDT3.



(e) IGD. ZDT6.



(f) IGD. ZDT6.

Figure 5.29.: Comparative study of ESPEA – convergence of IGD for problems ZDT2, ZDT3 and ZDT6. Straight lines indicate median performance and dashed lines IQRs of performance.

Table 5.10.: Comparative study of ESPEA – IGD. Median and IQR (as subscript) results after 50 000 function evaluations. Best performances are colored in green, second best in blue. ESPEA and SPEA2 are the top performing algorithms, while NSGA-III and SMPSO also achieve good results.

	ESPEA	IBEA	MOEAD	NSGA-II	NSGA-III	SMPSO	SMS-EMOA	SPEA2
B1	5.13e-4 <sub>1.1e-5</sub>	5.62e-4 <sub>2.6e-5</sub>	6.85e-4 <sub>3.6e-6</sub>	7.93e-4 <sub>1.0e-4</sub>	5.61e-4 <sub>3.0e-5</sub>	8.04e-4 <sub>9.3e-5</sub>	8.00e-4 <sub>1.1e-5</sub>	4.34e-4 <sub>9.1e-6</sub>
B2	4.64e-4 <sub>2.0e-5</sub>	6.47e-4 <sub>4.1e-5</sub>	6.38e-4 <sub>6.4e-6</sub>	1.02e-3 <sub>2.5e-4</sub>	5.76e-4 <sub>2.9e-5</sub>	8.77e-4 <sub>1.5e-4</sub>	7.04e-4 <sub>5.5e-7</sub>	3.60e-4 <sub>8.1e-6</sub>
B3	5.86e-4 <sub>1.2e-5</sub>	2.49e-3 <sub>2.8e-4</sub>	9.47e-4 <sub>1.1e-5</sub>	9.50e-4 <sub>1.6e-4</sub>	6.67e-4 <sub>4.8e-5</sub>	8.92e-4 <sub>1.1e-4</sub>	7.81e-4 <sub>1.7e-5</sub>	5.07e-4 <sub>1.2e-5</sub>
B4	6.17e-4 <sub>3.1e-5</sub>	5.02e-4 <sub>7.6e-6</sub>	7.73e-4 <sub>7.5e-6</sub>	1.29e-3 <sub>2.7e-4</sub>	5.27e-4 <sub>9.7e-6</sub>	9.83e-4 <sub>1.8e-4</sub>	7.72e-4 <sub>2.1e-5</sub>	4.62e-4 <sub>1.3e-5</sub>
B5	6.93e-4 <sub>2.0e-5</sub>	1.42e-3 <sub>7.2e-5</sub>	9.31e-4 <sub>1.0e-4</sub>	8.29e-4 <sub>5.6e-5</sub>	6.57e-4 <sub>3.0e-5</sub>	8.33e-4 <sub>6.3e-5</sub>	9.15e-4 <sub>1.7e-5</sub>	5.23e-4 <sub>1.1e-5</sub>
B6	7.38e-4 <sub>3.8e-5</sub>	1.66e-3 <sub>1.3e-4</sub>	8.33e-4 <sub>4.1e-5</sub>	8.55e-4 <sub>1.0e-4</sub>	6.75e-4 <sub>5.4e-5</sub>	8.05e-4 <sub>8.2e-5</sub>	1.09e-3 <sub>4.1e-5</sub>	5.26e-4 <sub>1.7e-5</sub>
DEB2DK $k = 1$	4.60e-5 <sub>4.3e-7</sub>	6.87e-5 <sub>3.2e-6</sub>	5.80e-5 <sub>6.4e-6</sub>	6.11e-5 <sub>5.8e-6</sub>	4.64e-5 <sub>9.4e-7</sub>	4.51e-5 <sub>3.9e-7</sub>	4.79e-5 <sub>8.4e-7</sub>	4.72e-5 <sub>8.9e-7</sub>
DEB2DK $k = 3$	4.57e-5 <sub>6.3e-7</sub>	3.74e-4 <sub>2.6e-5</sub>	5.67e-5 <sub>5.4e-6</sub>	5.97e-5 <sub>4.0e-6</sub>	4.54e-5 <sub>1.1e-6</sub>	4.47e-5 <sub>3.9e-7</sub>	4.80e-5 <sub>8.3e-7</sub>	4.65e-5 <sub>8.3e-7</sub>
DEB3DK $k = 1$	4.11e-4 <sub>1.0e-5</sub>	7.24e-4 <sub>1.5e-5</sub>	5.95e-4 <sub>5.9e-5</sub>	5.53e-4 <sub>9.2e-5</sub>	3.64e-4 <sub>1.3e-5</sub>	4.66e-4 <sub>3.8e-5</sub>	3.69e-4 <sub>7.6e-6</sub>	3.52e-4 <sub>1.4e-5</sub>
DEB3DK $k = 2$	4.27e-4 <sub>1.4e-5</sub>	9.40e-4 <sub>8.5e-6</sub>	6.41e-4 <sub>7.4e-5</sub>	5.26e-4 <sub>5.3e-5</sub>	3.69e-4 <sub>1.8e-5</sub>	4.70e-4 <sub>3.1e-5</sub>	4.75e-4 <sub>3.8e-5</sub>	3.62e-4 <sub>1.2e-5</sub>
DO2DK $k = 2, s = 1$	5.27e-5 <sub>7.5e-7</sub>	1.88e-3 <sub>6.5e-5</sub>	3.23e-4 <sub>4.5e-6</sub>	6.57e-5 <sub>4.1e-6</sub>	2.44e-4 <sub>3.3e-6</sub>	4.92e-5 <sub>1.8e-7</sub>	5.98e-5 <sub>1.7e-6</sub>	5.80e-5 <sub>1.7e-6</sub>
DO2DK $k = 4, s = 1$	5.26e-5 <sub>7.4e-7</sub>	9.37e-4 <sub>1.8e-5</sub>	3.40e-4 <sub>3.1e-6</sub>	6.54e-5 <sub>4.6e-6</sub>	2.59e-4 <sub>2.3e-6</sub>	4.94e-5 <sub>3.0e-7</sub>	7.51e-5 <sub>2.3e-6</sub>	5.78e-5 <sub>1.4e-6</sub>
DTLZ1	4.99e-4 <sub>6.6e-6</sub>	4.22e-3 <sub>4.5e-4</sub>	5.96e-4 <sub>7.3e-6</sub>	5.98e-4 <sub>3.9e-5</sub>	4.46e-4 <sub>2.1e-5</sub>	6.47e-4 <sub>4.8e-5</sub>	4.23e-4 <sub>4.6e-6</sub>	4.48e-4 <sub>1.2e-5</sub>
invDTLZ1	4.84e-4 <sub>6.5e-6</sub>	4.53e-3 <sub>5.8e-4</sub>	6.98e-4 <sub>7.2e-6</sub>	5.84e-4 <sub>4.4e-5</sub>	5.96e-4 <sub>2.0e-5</sub>	6.43e-4 <sub>4.7e-5</sub>	9.74e-4 <sub>2.6e-6</sub>	4.33e-4 <sub>1.1e-5</sub>
DTLZ3	7.20e-4 <sub>8.2e-5</sub>	5.31e-3 <sub>1.3e-4</sub>	7.97e-4 <sub>2.4e-4</sub>	8.42e-4 <sub>1.2e-4</sub>	6.69e-4 <sub>1.6e-4</sub>	7.84e-4 <sub>5.6e-5</sub>	8.41e-4 <sub>2.2e-5</sub>	7.56e-4 <sub>1.6e-3</sub>
DTLZ5	3.60e-4 <sub>2.4e-5</sub>	2.28e-3 <sub>1.4e-4</sub>	9.57e-4 <sub>5.9e-6</sub>	4.39e-4 <sub>3.3e-5</sub>	1.25e-3 <sub>2.7e-4</sub>	3.13e-4 <sub>1.2e-5</sub>	4.91e-4 <sub>3.2e-5</sub>	3.48e-4 <sub>1.4e-5</sub>
DTLZ7	4.13e-3 <sub>4.8e-3</sub>	5.18e-3 <sub>4.1e-3</sub>	1.24e-3 <sub>1.3e-4</sub>	5.98e-4 <sub>5.5e-5</sub>	5.66e-4 <sub>3.8e-5</sub>	7.27e-4 <sub>1.2e-4</sub>	9.02e-4 <sub>3.5e-3</sub>	5.03e-4 <sub>2.3e-5</sub>
Lamé $m = 2, \gamma = 0.25$	5.41e-5 <sub>2.2e-7</sub>	3.88e-3 <sub>4.2e-4</sub>	1.51e-3 <sub>2.2e-5</sub>	7.19e-5 <sub>4.5e-6</sub>	1.38e-3 <sub>6.7e-6</sub>	5.51e-5 <sub>5.0e-7</sub>	2.79e-4 <sub>1.0e-5</sub>	5.82e-5 <sub>1.5e-6</sub>
Lamé $m = 3, \gamma = 0.5$	1.45e-4 <sub>3.2e-6</sub>	2.74e-3 <sub>8.2e-4</sub>	2.92e-4 <sub>4.2e-6</sub>	2.07e-4 <sub>2.1e-5</sub>	4.27e-4 <sub>1.6e-4</sub>	1.81e-4 <sub>1.4e-5</sub>	2.08e-4 <sub>1.1e-5</sub>	1.56e-4 <sub>1.3e-5</sub>
Lamé $m = 2, \gamma = 4$	5.36e-5 <sub>8.0e-7</sub>	6.91e-4 <sub>6.6e-5</sub>	6.78e-5 <sub>5.0e-8</sub>	6.55e-5 <sub>4.3e-6</sub>	6.69e-5 <sub>5.2e-6</sub>	4.82e-5 <sub>2.9e-7</sub>	1.56e-4 <sub>8.4e-6</sub>	5.39e-5 <sub>1.5e-6</sub>
ZDT1	4.41e-5 <sub>4.5e-7</sub>	5.04e-5 <sub>1.7e-6</sub>	7.23e-5 <sub>4.8e-6</sub>	5.74e-5 <sub>3.2e-6</sub>	5.24e-5 <sub>1.8e-6</sub>	4.25e-5 <sub>2.9e-7</sub>	4.22e-5 <sub>2.4e-7</sub>	4.58e-5 <sub>1.2e-6</sub>
ZDT2	4.39e-5 <sub>4.5e-7</sub>	1.64e-4 <sub>1.0e-5</sub>	5.37e-5 <sub>3.1e-6</sub>	5.90e-5 <sub>3.5e-6</sub>	4.58e-5 <sub>1.3e-6</sub>	4.40e-5 <sub>5.6e-7</sub>	5.72e-5 <sub>3.0e-6</sub>	4.56e-5 <sub>6.9e-7</sub>
ZDT3	6.41e-5 <sub>1.2e-6</sub>	7.66e-4 <sub>2.6e-5</sub>	2.52e-4 <sub>7.3e-6</sub>	8.02e-5 <sub>4.6e-6</sub>	9.24e-5 <sub>9.0e-6</sub>	6.14e-5 <sub>6.9e-7</sub>	6.28e-5 <sub>2.3e-6</sub>	7.22e-5 <sub>3.4e-6</sub>
ZDT6	3.23e-5 <sub>9.0e-7</sub>	7.16e-5 <sub>6.0e-6</sub>	3.61e-5 <sub>6.0e-8</sub>	5.46e-5 <sub>8.9e-6</sub>	3.34e-5 <sub>1.3e-6</sub>	3.68e-5 <sub>3.1e-6</sub>	3.63e-5 <sub>1.1e-6</sub>	3.62e-5 <sub>2.7e-6</sub>

Table 5.11.: Comparative study of ESPEA – IGD. The table shows  $p$ -values of a post-hoc analysis. Green cell color indicates that ESPEA outperforms the corresponding algorithm with confidence at a 95 % level, the blue color without confidence. Red cell color indicates that ESPEA is outperformed by the corresponding algorithm with confidence at a 95 % level, the orange color without confidence. ESPEA outperforms most of the other algorithms with confidence on the different problem instances.

	IBEA	MOEAD	NSGA-II	NSGA-III	SMPSO	SMS-EMOA	SPEA2
B1	0.0002	0.0000	0.0000	0.0001	0.0000	0.0000	0.0560
B2	0.0000	0.0000	0.0000	0.0522	0.0000	0.0000	0.0601
B3	0.0000	0.0000	0.0000	0.0493	0.0000	0.0000	0.0583
B4	0.0000	0.0000	0.0000	0.0558	0.0000	0.0004	0.0000
B5	0.0000	0.0000	0.0000	0.4458	0.0000	0.0000	0.0000
B6	0.0000	0.0000	0.0000	0.1184	0.0002	0.0000	0.0000
DEB2DK $k = 1$	0.0000	0.0000	0.0000	0.5216	0.0034	0.0000	0.0000
DEB2DK $k = 3$	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000	0.0856
DEB3DK $k = 1$	0.0000	0.0000	0.0000	0.0000	0.0292	0.0005	0.0000
DEB2DK $k = 2$	0.0000	0.0000	0.0000	0.0009	0.0006	0.0000	0.0000
DO2DK $k = 2$ $s = 1$	0.0000	0.0000	0.0000	0.0000	0.0602	0.0000	0.0134
DO2DK $4 = 2$ $s = 1$	0.0000	0.0000	0.0000	0.0000	0.0615	0.0000	0.0577
DTLZ1	0.0000	0.0001	0.0000	0.0012	0.0000	0.0000	0.0013
DTLZ3	0.0000	0.0000	0.0000	1.0000	0.0001	0.0000	0.0000
DTLZ5	0.0000	0.0000	0.0013	0.0000	0.0000	0.0000	0.9806
DTLZ7	0.0000	0.0009	0.0001	0.0000	0.9108	0.1354	0.0000
invDTLZ1	0.0000	0.0000	0.0001	0.0000	0.0000	0.0000	0.0602
Lamé $m = 2$ $\gamma = 0.25$	0.0000	0.0000	0.0000	0.0000	0.0630	0.0000	0.0000
Lamé $m = 3$ $\gamma = 0.5$	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.1536
Lamé $m = 2$ $\gamma = 4$	0.0000	0.0000	0.0000	0.0000	0.0004	0.0000	1.0000
ZDT1	0.0000	0.0000	0.0000	0.0000	0.0124	0.0000	0.0784
ZDT2	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000
ZDT3	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000	0.0074
ZDT6	0.0000	0.0000	0.0000	0.0001	0.0000	0.0000	0.0000

### 5.3. Computational Analysis of Preference-biased Approximations

The next part of the analysis focuses on the ranking of algorithms across the entire set of optimization problems in the study. In order to facilitate a meaningful basis for comparison the following procedure is proposed. Each algorithm is assigned a performance rank on every test problem consisting of the number of other algorithms it outperforms at a 5% significance level. A rank of 7, for example, indicates that the given algorithm outperforms all other algorithms on the respective problem. These performance ranks are then used as input data in a Friedman test [Fri37] to establish an overall ranking of the algorithms across all problems (see Table G.21).

The Friedman test is a non-parametric test that is similar to the Kruskal-Wallis test. The null hypothesis of the Friedman test also states that all considered samples – performance ranks of an algorithm across all test problems – stem from the same distribution implying that observed performance differences are only a product of chance. No assumption about the shape of the underlying distribution is made. In contrast to the Kruskal-Wallis test, the Friedman test assumes that the observations in the data samples are paired – ranks are considered within and not across problems.

The proposed procedure explicitly refrains from using median or mean IGD, since they are no indicator for significant performance difference. A given algorithm can only be considered superior to another algorithm on a given test problem if a statistical test rules out that the observed performance difference may be attributed to chance.

The Friedman test reported a  $p$ -value of  $6.6183e-16$  strongly indicating that there exist significant performance differences. A post-hoc analysis using Dunn's method was performed to assess between which algorithms significant performance differences occur. The results show that ESPEA outperforms IBEA, MOEAD, NSGA-II and SMS-EMOA with high confidence. ESPEA achieves better results than SMPSO and NSGA-III, however the observed performance differences are not significant. SPEA2 is the only algorithm that reaches a higher performance rank than ESPEA, however the ranks of both algorithms differ only marginally. A  $p$ -value of 1.0 suggests that virtually no performance difference exists.

The observed performance differences related to the problem characteristics are analyzed next using statistical methods. The influence of the number of objectives on the algorithms' performance ranks is considered first. The problems were divided into two groups of two and three objective problems. A Wilcoxon rank sum test [MW47] was used to compare both groups. The Wilcoxon rank sum test is a non-parametric test, whose null hypothesis states that the medians of both groups are the same. If the null hypothesis is rejected the respective algorithm exhibits a significant performance difference between two and three objective problems in comparison to the other algorithms of the experiment. If

## 5. Preference-biased Pareto Front Approximations

Table 5.12.: Comparative study of ESPEA – IGD. Post-hoc analysis of the Friedman test results. The rank states the number of other algorithms an algorithm significantly outperforms on average. The column  $p$ -values is related to the performance difference with respect to ESPEA. Green cell color indicates that ESPEA outperforms the corresponding algorithm with confidence at a 95 % level, the blue color without confidence. Orange cell color indicates that ESPEA is outperformed without confidence. ESPEA is second best performing algorithm and no significant performance difference is observed between ESPEA and the top performing SPEA2.

Rank	Algorithm	$p$ -value
5.6250	SPEA2	1.0000
5.5417	ESPEA	n/a
4.4375	SMPSO	0.9622
4.1250	NSGA-III	0.6862
3.0417	SMS-EMOA	0.0084
2.3750	NSGA-II	0.0001
2.0000	MOEAD	0.0000
0.8542	IBEA	0.0000

the performance differences are not significant the performance ranks are similar for two and three objectives.

Table 5.13 shows the results of the Wilcoxon rank sum test. The quantitative comparison confirms that MOEAD and NSGA-III perform better on three objective problems as expected, however only NSGA-III exhibits a significant performance increase in comparison to the other algorithms. ESPEA performs slightly worse on three objective problems, however the difference is not significant. As speculated, SMPSO shows a large performance drop, which is confirmed to be significant by the rank sum test. SPEA2, on the hand, is found to achieve significantly better results on three objective problems. This observation, however, does not imply that SPEA2's niching mechanism performs better on three compared to two objective problems in general, since only the relative performance between algorithms is compared. Instead, the results rather suggest that other niching mechanisms achieve more uniform Pareto front approximations on two objective problems and that SPEA2's niching mechanism does not exhibit similar performance drops observed in other MOEAs if the number of objectives is increased from two to three.

The analysis of the computational study is completed by assessing the results of the second experiment. Figure 5.30 shows the convergence plots for the IGD metric on the three test distances. The majority of the algorithms exhibit a

### 5.3. Computational Analysis of Preference-biased Approximations

Table 5.13.: Comparative study of ESPEA – IGD. Comparison with respect to the number of objectives. Problems are grouped into two and three objective problems and compared by means of a Wilcoxon rank sum test. Significant performance differences between two and three objective problems are colored in green. The rank states the average number of other algorithms the given algorithm significantly outperforms for the stated number of objectives. Performance differences are significant for NSGA-III, SMPSO and SPEA2.

	ESPEA	IBEA	MOEAD	NSGA-II
<i>p</i> -value	0.6024	0.3727	0.0734	0.8981
Rank $m = 2$	5.3000	0.2000	1.0000	1.7000
Rank $m = 3$	4.8462	0.9231	1.6923	1.6923
	NSGA-III	SMPSO	SMS-EMOA	SPEA2
<i>p</i> -value	0.0128	3.44e-4	0.2089	4.82e-4
Rank $m = 2$	2.5000	5.9000	3.0000	3.7000
Rank $m = 3$	4.5385	2.0000	2.1538	5.8462

similar performance quickly converging towards the Pareto front and showing no sign of further improvement after 50 to 100 generations. IBEA, however, performs notably worse than the other algorithms on all test instances. Its hypervolume-based niching mechanism does not appear to be able to handle extreme curvatures. Since IBEA does not consider the overall hypervolume of the entire population (in contrast to SMS-EMOA), but only pairwise hypervolume comparisons between population members, strong curvatures might lead IBEA to overemphasize knee points and neglect other regions of the Pareto front. MOEAD and NSGA-III perform considerably worse on the convex test instances than the other algorithms. An explanation for this behavior is explored later when the results with respect to the spread metric are additionally taken into account.

Median IGD and IQRs are displayed in Table 5.14. In contrast to the convergence analysis, the figures suggest that all algorithms achieve good results on the convex, concave and mixed curvature problems. The IGD obtained by the algorithms is of the same magnitude ( $1e-4$ ) on all problems with the exception of IBEA on the concave problems. ESPEA and SMPSO obtain the smallest IGDs among all algorithms.

A post-hoc analysis is performed to check the performance differences for significance. A Kruskal-Wallis test confirms that significant performance differences exist on all three problem instances (Table G.16). Table 5.15 shows the results of the post-hoc analysis. ESPEA outperforms IBEA, MOEAD, NSGA-

## 5. Preference-biased Pareto Front Approximations

Table 5.14.: Comparative study of ESPEA – IGD. Median and IQR (as subscript) results after 50 000 function evaluations grouped by the curvature of the Pareto front. Best performances colored in green, second best in blue. Significant performance differences are observed on all problem instances. ESPEA and SMPSO are the top performing algorithms.

	ESPEA	IBEA	MOEAD	NSGA-II
Convex	1.32e-4 <sub>2.45e-5</sub>	4.74e-4 <sub>4.73e-3</sub>	4.21e-4 <sub>1.12e-3</sub>	1.89e-4 <sub>3.37e-5</sub>
Concave	1.53e-4 <sub>1.95e-5</sub>	1.09e-3 <sub>9.65e-4</sub>	1.83e-4 <sub>3.26e-5</sub>	1.99e-4 <sub>1.90e-5</sub>
Mixed	1.45e-4 <sub>2.00e-6</sub>	9.09e-4 <sub>2.13e-4</sub>	1.84e-4 <sub>1.95e-5</sub>	1.89e-4 <sub>1.59e-5</sub>
	NSGA-III	SMPSO	SMS-EMOA	SPEA2
Convex	3.48e-4 <sub>1.15e-3</sub>	1.44e-4 <sub>3.07e-5</sub>	1.57e-4 <sub>1.80e-4</sub>	1.45e-4 <sub>2.32e-5</sub>
Concave	1.74e-4 <sub>3.60e-5</sub>	1.51e-4 <sub>7.34e-6</sub>	2.83e-4 <sub>1.85e-4</sub>	1.58e-4 <sub>1.47e-5</sub>
Mixed	1.46e-4 <sub>3.71e-6</sub>	1.41e-4 <sub>1.15e-6</sub>	1.52e-4 <sub>3.32e-6</sub>	1.48e-4 <sub>2.95e-6</sub>

Table 5.15.: Comparative study of ESPEA – IGD. The table shows  $p$ -values of the post-hoc analysis based on the grouping by Pareto front curvature. Green cell color indicates that ESPEA outperforms the corresponding algorithm with confidence at a 95 % level, the blue color without confidence. Red cell color indicates that ESPEA is outperformed by the corresponding algorithm with confidence at a 95 % level, the orange color without confidence. ESPEA outperforms most of the other algorithms across the three test instances.

	IBEA	MOEAD	NSGA-II	NSGA-III	SMPSO	SMS-EMOA	SPEA2
Convex	0.0000	0.0000	0.0000	0.0000	0.3359	0.0000	0.0114
Concave	0.0000	0.0000	0.0000	0.0000	0.9600	0.0000	0.9962
Mixed	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000	0.0605

II and SMS-EMOA with confidence on all test instances. Furthermore, ESPEA achieves better results than NSGA-III and SPEA2, however only the difference on the convex instance is found to be significant for both algorithms. ESPEA is solely outperformed by SMPSO on the concave and mixed curvature problems. Additionally, the mixed curvature problems are the only test instance on which ESPEA performs significantly worse than any other algorithm.

Absolute IGDs after 50 000 function evaluations suggest that the algorithm performances are similar and that the obtained Pareto front approximations are close and well-spread across the true Pareto front. For this reason, the spread metric is considered to assess solely the uniformity of the distribution of points across the Pareto front (Table 5.16). In contrast to IGD, the spread metric reveals considerable performance differences. ESPEA and SMPSO achieve again the best results, however they also outperform most of the other algorithms by a large margin.

### 5.3. Computational Analysis of Preference-biased Approximations

Table 5.16.: Comparative study of ESPEA – spread. Median and IQR (as subscript) results after 50 000 function evaluations grouped by the curvature of the Pareto front. Best performances colored in green, second best in blue. ESPEA and SMPSO are the top performing algorithms.

	ESPEA	IBEA	MOEAD	NSGA-II
Convex	0.07 <sub>0.01</sub>	0.34 <sub>0.42</sub>	0.52 <sub>0.52</sub>	0.37 <sub>0.04</sub>
Concave	0.07 <sub>0.02</sub>	0.85 <sub>0.47</sub>	0.26 <sub>0.01</sub>	0.38 <sub>0.04</sub>
Mixed	0.09 <sub>0.01</sub>	0.90 <sub>0.38</sub>	0.32 <sub>0.03</sub>	0.36 <sub>0.04</sub>
	NSGA-III	SMPSO	SMS-EMOA	SPEA2
Convex	0.44 <sub>0.67</sub>	0.07 <sub>0.06</sub>	0.11 <sub>0.35</sub>	0.15 <sub>0.02</sub>
Concave	0.24 <sub>0.07</sub>	0.13 <sub>0.06</sub>	0.39 <sub>0.31</sub>	0.14 <sub>0.02</sub>
Mixed	0.15 <sub>0.02</sub>	0.07 <sub>0.01</sub>	0.15 <sub>0.02</sub>	0.15 <sub>0.02</sub>

The results in Table 5.16 additionally confirm many of the observations made for the analysis of the results in Table 5.10 Hypervolume-based methods such as IBEA and SMS-EMOA perform worse on concave Pareto fronts, since points towards the boundary contribute less to the overall hypervolume of the population on concave fronts. IBEA performs worst on the mixed curvature instance. This may be attributed to IBEA failing to approximate the concave bulges of the front (Figure 5.31). The reference point based methods MOEAD and NSGA-III perform worst on the convex fronts and best on the mixed curvature fronts. The reason for this can be explained as follows: The niching mechanisms of both algorithms aim at finding points that lie at the intersection of the line passing through the reference point and the origin with the Pareto front. Reference points that are uniform on the unit simplex<sup>7</sup> do not produce uniform intersections on the Pareto front (Figure 5.31). Since the Pareto front of the mixed curvature instances are close to the unit simplex, MOEAD and NSGA-III perform considerably better on these problems compared to the convex instances.

Finally, NSGA-II and SMPSO exhibit vast performance differences although both algorithms use crowding distance as niching mechanism. The performance difference can be explained by the steady-state update mechanism of the archive employed by SMPSO. NSGA-II computes the crowding distance for all  $2N$  population members of parents and offspring if all population members are nondominated to each other. The  $N$  candidate solutions possessing the smallest crowding distances based on the entire population survive to the next round. This subset of the population, however, is not necessarily the subset that minimizes the sum of all crowding distances if crowding distances are only computed based on the elements of the subset and not the entire population of  $2N$  elements. SMPSO, on the other hand, recomputes crowding distances

<sup>7</sup>In case of two objectives the unit simplex consists of the line between (1, 0) and (0, 1).

## 5. Preference-biased Pareto Front Approximations

Table 5.17.: Comparative study of ESPEA – spread. The table shows  $p$ -values of the post-hoc analysis based on the grouping by Pareto front curvature. Green cell color indicates that ESPEA outperforms the corresponding algorithm with confidence at a 95 % level, the blue color without confidence. Orange cell color indicates that ESPEA is outperformed by the corresponding algorithm without confidence at a 95 % level. ESPEA outperforms most of the other algorithms across the three test instances.

	IBEA	MOEAD	NSGA-II	NSGA-III	SMPSO	SMS-EMOA	SPEA2
Convex	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000
Concave	0.0000	0.0000	0.0000	0.0000	0.0532	0.0000	0.0000
Mixed	0.0000	0.0000	0.0000	0.0000	0.1185	0.0000	0.0000

whenever a candidate solution is added to the archive. Thereby, the element that contributes least to the overall crowding distance of the archive is eliminated. Therefore, SMPSO is able to achieve much uniformer approximations of the Pareto front.

Table 5.17 lists the results of a post-hoc analysis for the spread metric (see Table G.17 for the corresponding Kruskal-Wallis test results). The figures show that ESPEA outperforms all algorithms besides SMPSO with confidence. ESPEA is outperformed by SMPSO on the convex and mixed fronts, however, the performance difference is not significant.

The computational study has revealed that ESPEA is a highly competitive algorithm for solving MOOPs. ESPEA outperforms many algorithms – some even by a large margin – that are widely used in research and in practice. At the same time, there exists no algorithm that obtains clearly better results than ESPEA. ESPEA, therefore, constitutes an excellent choice for obtaining uniform finite set of points approximations.

### 5.3. Computational Analysis of Preference-biased Approximations

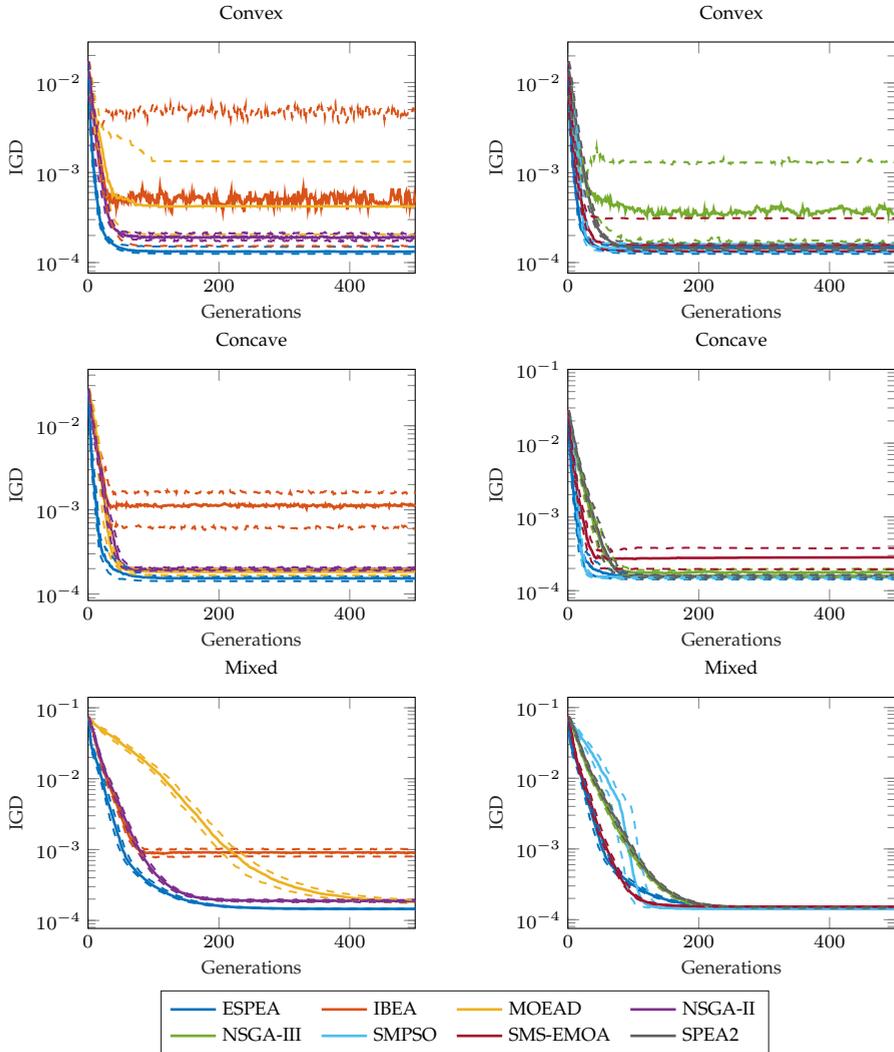


Figure 5.30.: Comparative study of ESPEA – convergence of IGD for convex, concave and mixed curvature problems. Straight lines indicate median performance and dashed lines IQRs of performance.

## 5. Preference-biased Pareto Front Approximations

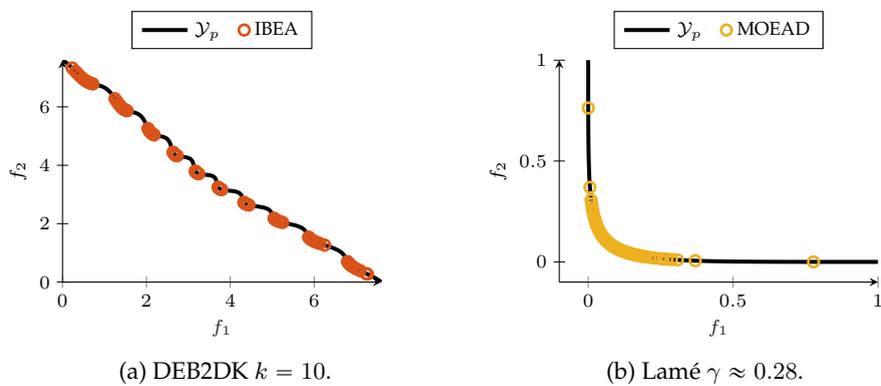


Figure 5.31.: Example runs of IBEA and MOEAD on DEB2DK with  $k = 10$  and Lamé with  $\gamma \approx 0.28$ . The niching mechanisms of both algorithms are not able to produce uniform Pareto front approximations.

### 5.3.3. Qualitative Analysis of Preference-biased Approximations

The final computational study of this chapter focuses on analyzing preference-biased Pareto front approximations generated by ESPEA using scalarized preference information. The analysis is purely qualitative and of illustrative nature to underline the usefulness of such approximations to a DM. ESPEA uses the same search operator configuration as in the previous study in Section 5.3.1 (Table E.2). Approximations of required reference points for computing scalarization values were estimated based on the current population. ESPEA was run on each problem for 20 000 function evaluations with the exception of B3 and DTLZ3, for which 25 000 and 50 000 function evaluations were used, respectively. A population size of 50 was chosen for bi-objective problems and a population size of 100 for problems featuring three objectives.

Figures 5.32 to 5.35 show exemplary runs of ESPEA on three-objective problems using different scalarization functions for computing charges. The surfaces of the Pareto fronts are colored to depict the desirable regions identified by the respective scalarization function and called scalarization landscape or also scalarization surface for three objectives. Figures 5.36 to 5.40 depict ESPEA runs on bi-objective problems. Each figure features a plot of the corresponding scalarization function that was used. The plot indicates the scalarization value that the function assigns to the corresponding Pareto optimal objective vector on the  $f_1$ -axis. The function  $\Psi := 1$  refers to the case in which all solutions are equally desirable – also called the no preference case. The identifiers of the other scalarization functions are contained in Table 3.1.

Figure 5.32 shows example runs of ESPEA on DTLZ3 and DEB3DK ( $k = 2$ ) using the Chebyshev and sum of objectives scalarization functions, respectively. The scalarization surface of DTLZ3 using Chebyshev’s method shows that the most desirable solutions are located around the section of the median lines to the centroid of the front. The three lines form a distinctive cross as indicated by the blue shading. In the no preference case, the solutions are uniformly spread across the front. If the Chebyshev method is utilized, the Pareto front approximation prominently identifies the cross as desirable region and locates more solutions around the three intersecting lines.

A similar observation can be made in Figure 5.32 for DEB3DK with  $k = 2$ . The solutions are distributed uniformly across the Pareto front avoiding the holes in the surface in the no preference run. According to the sum of objectives, the most desirable solutions are located at the convex bulges next to the two large holes at the top of the front and the convex bulges near the barycenter of the front. ESPEA focuses more solutions in exactly these regions if the sum of objectives is chosen as scalarization function.

## 5. Preference-biased Pareto Front Approximations

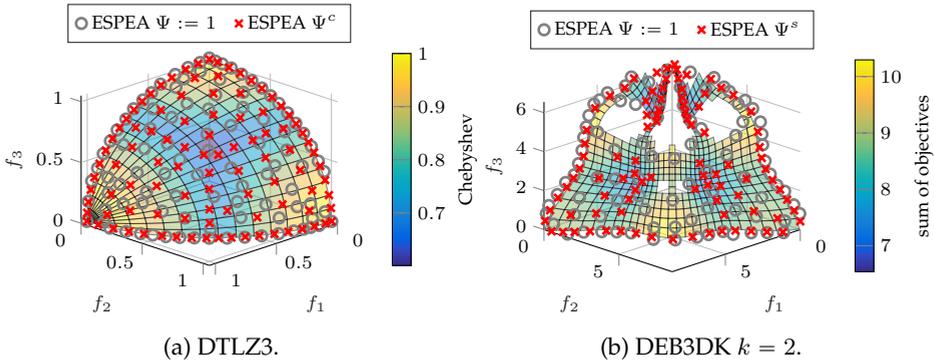


Figure 5.32.: ESPEA runs on DTLZ3 using the Chebyshev method and on DEB3DK ( $k = 2$ ) using the sum of objectives.

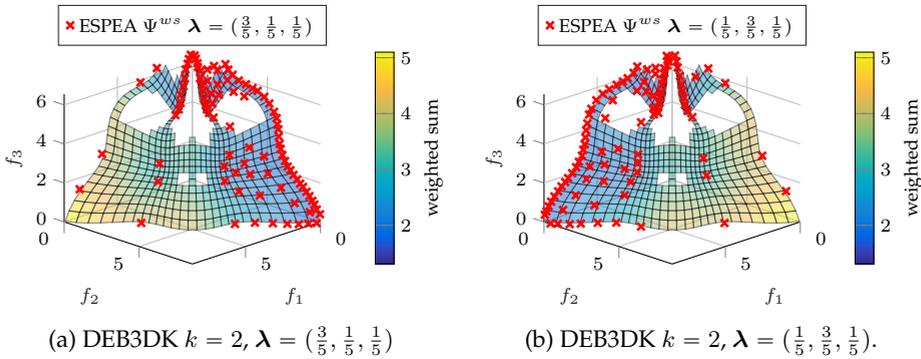


Figure 5.33.: ESPEA runs on DEB3DK ( $k = 2$ ) using the weighted sum.

The weighted sum method allows giving different priorities to the objective functions that are optimized. Prioritizing objectives leads to large shifts in the regions of the Pareto front that are deemed most desirable. This shift in priorities is also reflected in the preference-biased approximation obtained by ESPEA. Figure 5.33 shows the influence of different weights on ESPEA's search results. Prioritizing either objective one or two over the other objectives leads to a strong concentration of solutions that feature smaller  $f_1$ - and  $f_2$ -values respectively. Both Pareto front approximations are nearly symmetric to each other providing further evidence that  $N$ -optimal distributions are also obtained beyond the no preference case.

Figure 5.34 shows runs of ESPEA on DTLZ7 and DEB3DK with  $k = 1$  using

### 5.3. Computational Analysis of Preference-biased Approximations

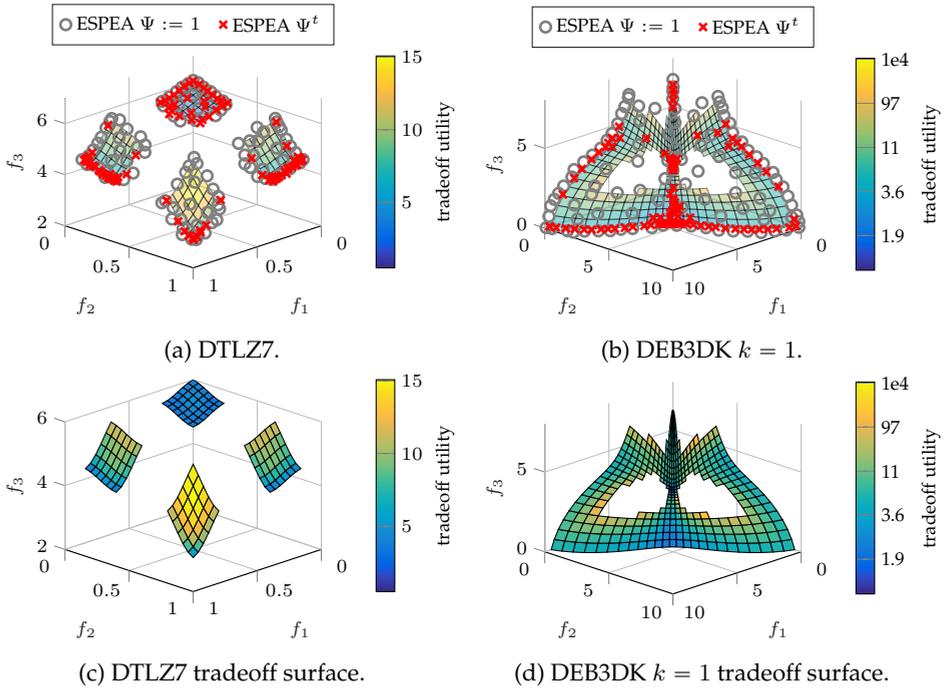


Figure 5.34.: ESPEA runs on DTLZ7 and DEB3DK ( $k = 1$ ) using tradeoff utility.

tradeoff utility as scalarization function. The figure additionally contains two plots that show only the scalarization surfaces to better illustrate the desirable regions of the Pareto fronts. The northern patch of DTLZ7 is deemed most interesting, whereas the other three patches exhibit desirable regions towards their southern boundary. In the no preference case, solutions are distributed uniformly across the four patches. If tradeoff utility is chosen as scalarization function, ESPEA strongly concentrates solutions on those parts of the patches that feature the smallest tradeoff utility. The approximation of the northern patch is less dense towards the center. This can be attributed to the solutions on the boundary of the patch being located further away from solutions on the same patch. Thereby they introduce less energy into the archive.

The Pareto front of DEB3DK exhibits both strong convex and concave curvatures leading to tradeoff utility becoming quite large in areas close to the holes that are nearly Pareto dominated. This in turn leads to solutions being highly concentrated in areas that feature a very small tradeoff utility. Regions that possess a larger tradeoff utility are nearly universally neglected. Using trade-

### 5. Preference-biased Pareto Front Approximations

off utility as scalarization function in ESPEA even appears to speed up the convergence towards the Pareto front on DEB3DK.

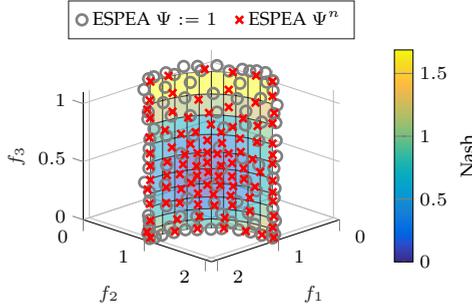


Figure 5.35.: ESPEA runs on B3 using the Nash bargaining solution.

Figure 5.35 shows a run of ESPEA on B3 using the modified Nash bargaining solution as scalarization function. The most desirable region of the front according to the Nash bargaining solution is located at the center of the front slightly shifted towards the minimum of  $f_3$ . In the no preference case, ESPEA distributes the solutions evenly across the front. Using the Nash bargaining solution, more solutions are located towards the center. These results also suggest that ESPEA is able to correctly estimate the Nadir point during runtime, which is necessary to correctly compute the Nash bargaining solution.

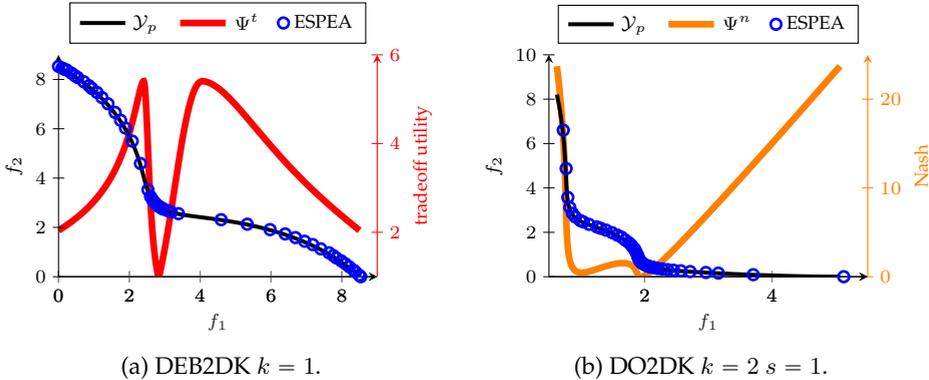


Figure 5.36.: ESPEA runs on DEB2DK ( $k = 1$ ) using tradeoff utility and on DO2DK ( $k = 2, s = 1$ ) using the Nash bargaining solution.

Tradeoff utility on DEB2DK  $k = 1$  possesses a global optimum at the center of the front that is enclosed in a narrow valley resembling a gorge (Figure 5.36).

### 5.3. Computational Analysis of Preference-biased Approximations

The curvature of the scalarization landscape becomes more flat towards the boundary of the Pareto front with two local optima being located at the extreme points. ESPEA's Pareto front approximation focuses strongly on the knee region around the global tradeoff optimum and, at the same time, avoids the areas possessing large tradeoff utility that encompass the knee region. The approximation also becomes more dense towards the boundary when tradeoff utility decreases.

The Nash bargaining landscape on DO2DK features a large valley that exhibits a strong increase in scalarization values to the left extreme point and a moderate increase towards the right extreme point. ESPEA focuses the vast majority of the solutions at the bottom of the valley. Only a few solutions are found towards the boundary yielding enough information for the DM to allow a correct estimation of the course of the front.

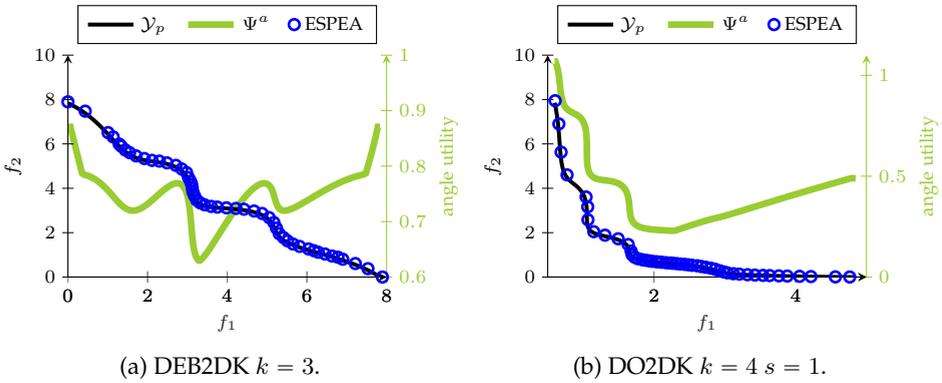


Figure 5.37.: ESPEA runs on DEB2DK ( $k = 3$ ) and DO2DK ( $k = 4$ ,  $s = 1$ ) using angle utility.

Example runs of ESPEA using angle utility are depicted in Figure 5.37. The angle utility landscape of DEB2DK ( $k = 3$ ) features a pronounced valley at the center of the front and is surrounded by two smaller valleys. Most of the solutions are focused in the large valley, however the two smaller valleys are also well approximated. On DO2DK ( $k = 4$ ,  $s = 1$ ), the angle utility landscape is rather flat towards the right extreme point and takes a terrace-like shape when moving to the left extreme point. The majority of the solutions are located in the valley basin and the flat ascent towards the right extreme point. Only few solutions are found on the terrace, which features a steep increase in angle utility.

Two instances of the Lamé problem with  $m = 2$  possessing extreme curvatures are considered in Figure 5.38. The sum of objectives identifies the most

## 5. Preference-biased Pareto Front Approximations

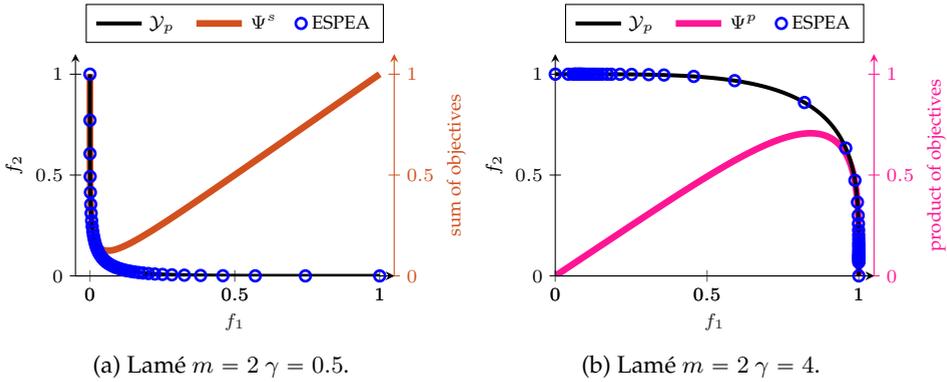


Figure 5.38.: ESPEA runs on Lamé ( $m = 2, \gamma = 4$ ) with the Chebyshev method and on Lamé ( $m = 2, \gamma = 0.25$ ) with the sum of objectives.

desirable region at the strong convex bulge for  $\gamma = 0.5$ . ESPEA generates an approximation that appears to be symmetrical to the bisecting line of the first quadrant. Such a behavior is expected, since the Pareto front is symmetrical to the bisecting line and any point on the front and its mirror image possess the same sum of objectives. This observation further suggests that ESPEA is able to approximate  $N$ -optimal distributions of points for arbitrary scalarization functions.

For  $\gamma = 4$ , the product of objectives identifies the extreme points as most desirable options on the Lamé problem instance. A global optimization of the scalarization function would only yield either extreme point as solution. ESPEA, on the other hand, generates more options such that the entire Pareto front is covered by points. The distribution of points is more dense towards the extreme points, where the product of objectives is smallest.

Runs of ESPEA on ZDT1 and ZDT2 using the Chebyshev method and tradeoff utility, respectively, are depicted in Figure 5.39. The scalarization landscape of the Chebyshev method resembles a wedge exhibiting a shallow curvature from the global optimum towards the extreme points. The density of points steadily increases towards the global optimum. The scalarization landscape of ZDT2 is discontinuous at the extreme points, since the Pareto front of ZDT2 is concave. The global scalarization optima are located at the extreme points. The trade-off landscape features a shallow valley that extends towards the right extreme point and possesses a tangential tradeoff towards the left extreme points. ESPEA identifies both global optima and densely approximates the shallow valley while neglecting the area of steep ascent towards the left extreme point.

### 5.3. Computational Analysis of Preference-biased Approximations

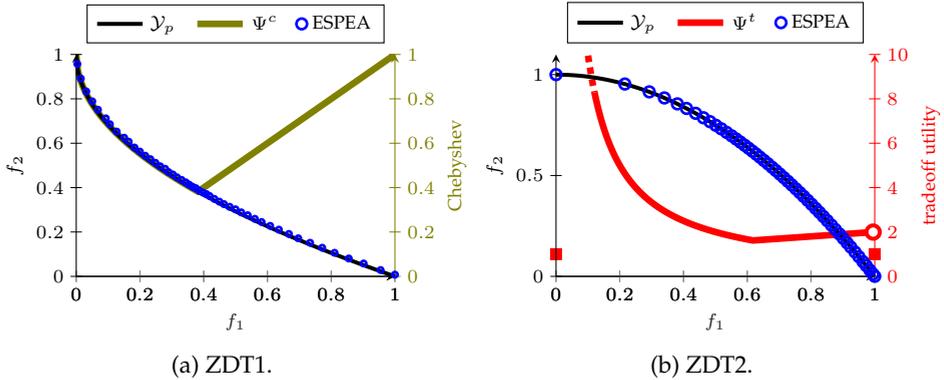


Figure 5.39.: ESPEA runs on ZDT1 using the Chebyshev method and on ZDT2 using tradeoff utility.

As suggested in Section 5.1, the Pareto dominance relation that is used to update the archive in Lines 5 and 6 of Algorithm 1 may be substituted by any other dominance relation. The last two example runs in Figure 5.40 show ESPEA using the weighted sum in conjunction with tradeoff domination (Definition 16). No solutions are located in tradeoff dominated areas of the front even if the weighted sum identifies them as highly desirable.

The analysis has shown that ESPEA is able to generate preference-biased Pareto front approximations that takes scalarized preference information into account. ESPEA computes Pareto front approximations that concentrate more solution in areas that feature small scalarization values and only sparsely approximates regions possessing large scalarization values. The density of solutions on the front is strongly influenced by the composition of the scalarization landscape. If a scalarization landscape features gorges then most of the solutions are located at the valley floors. In contrast, if the landscape is flat solutions are more equally distributed across the fronts. It was also shown that ESPEA can be successfully combined with dominance notions beyond Pareto optimality.

## 5. Preference-biased Pareto Front Approximations

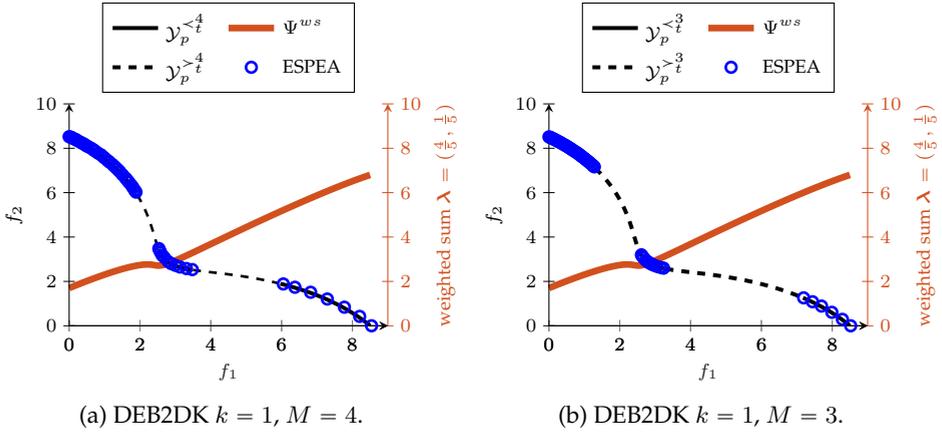


Figure 5.40.: ESPEA runs on DEB2DK  $k = 1$  using a weighted sum with  $\lambda = (\frac{4}{5}, \frac{1}{5})$  and tradeoff domination with  $M = 4$  and  $M = 3$ , respectively. The set  $\mathcal{Y}_p^{<M} := \{\mathbf{u} \in \mathcal{Y}_p \mid \nexists \mathbf{v} \in \mathcal{Y} : \mathbf{v} <_t^M \mathbf{u}\}$  is a subset of the Pareto front that consists of all tradeoff optimal points for a given  $M$ . Conversely,  $\mathcal{Y}_p^{>M} := \{\mathbf{u} \in \mathcal{Y}_p \mid \exists \mathbf{v} \in \mathcal{Y} : \mathbf{v} <_t^M \mathbf{u}\}$  consists of all tradeoff dominated points of the Pareto front for a given  $M$ .

## 6. Multimodal Scalarized Preferences

*Declaration: Parts of this chapter have been published in [BHSS17].*

Scalarized preferences invoke a desirability landscape on the Pareto front that may consist of multiple hills and valleys. Recall that scalarization functions are minimized within this work. The peak of any hill is a locally least desirable point and the bottom of a valley is a local desirability optimum. Any such optimum is an optimal choice in its immediate neighborhood given the scalarized preference. As explained in Section 4.1, the set of scalarization optima is expected to be much smaller than a finite set of points or subset approximation of the Pareto front. A DM that can focus only on the scalarization optima and should therefore be enabled to arrive quicker at a compromise solution that suits his preferences best.

The first part of this chapter presents the theoretical foundation of scalarization optima. A mathematical definition of local scalarization optima is given and the notion of basins of attraction is presented – a concept that is utilized in devising search strategies for local scalarization optima. Next, an algorithmic framework is presented for approximating the local scalarization optima of generic MOOPs and arbitrary scalarization functions. The final section consists of a computational study that analyzes the proposed approach on a set of artificial benchmark problems. The analysis is focused on identifying optimal components for each step of the algorithm. The results reveal that the approach is successful in approximating local scalarization optima on the considered problems and scalarization functions.

### 6.1. Local Scalarization Optima and Basins

Optimizing scalarized MOOPs usually yields only a single global optimum. The identified optimum, however, is not necessarily the best choice, if the scalarization function is not a perfect representation of the DM's preferences. In practice, it is often impractical or even infeasible to find a scalarization function that is a perfect representation of the DM's preferences [Mie99, Sec. 4.1]. In

## 6. Multimodal Scalarized Preferences

the presence of multiple global optima, the DM might prefer some optima to others because of hidden features that cannot be captured by the scalarization function. Such preferences, for example, may depend on the composition of the corresponding decision vectors of the optima. Global optimization algorithms, however, usually obtain only either of these global optima prohibiting the DM to make her own choice. In a different scenario, the global optimum might be counterintuitive to the preferences that the DM stated. As mentioned before, a weighted sum, for example, finds only extreme points on concave Pareto fronts. Any extreme point would be undesirable if the DM desires a solution that strikes a balance between the different objectives. Finally, the global optimum that was found might prove difficult to implement because of technical limitations in the manufacturing processes that cannot be anticipated beforehand. In all these cases, obtaining and considering additional global and local optima may enable the DM to make a better choice [Pre15].

### 6.1.1. Local Scalarization Optima

In order to formally define local scalarization optima, the concept of neighborhoods needs to be introduced first. Given an  $\varepsilon > 0$ , a neighborhood of a point  $\mathbf{u} \in \mathcal{Y}$  is a subset of the objective space that consists of all feasible objective vectors that lie within a distance  $\varepsilon$  of  $\mathbf{u}$  according to some metric, e.g. the Euclidean norm. A point  $\mathbf{u} \in \mathcal{Y}_p$  then qualifies as local scalarization optimum if there exists a non-empty neighborhood of  $\mathbf{u}$  such that  $\mathbf{u}$  possesses the smallest scalarization value among all Pareto optimal elements in the neighborhood.

**Definition 59** (local scalarization optimum [BHSS17]). *Let  $f$  be the objective function of an MOOP,  $\Psi$  be a scalarization function and for any  $\mathbf{u} \in \mathbb{R}^m$  let  $B(\mathbf{u}) = \{\mathbf{v} \in \mathbb{R}^m \mid \|\mathbf{u} - \mathbf{v}\|_2 < \varepsilon\}$  with  $\varepsilon > 0$  denote the open ball of  $\mathbf{u}$  with radius  $\varepsilon$ . An objective vector  $\mathbf{u}^L \in \mathcal{Y}_p$  is called a local scalarization optimum of  $\Psi$  on  $f$  if there exists a non-empty neighborhood of Pareto optimal points  $V(\mathbf{u}^L) = B(\mathbf{u}^L) \cap \mathcal{Y}_p$  with  $\varepsilon > 0$  such that for all  $\mathbf{v} \in V(\mathbf{u}^L)$  it holds that  $\Psi(\mathbf{u}^L) < \Psi(\mathbf{v})$ .*

- Any  $\mathbf{x}^L$  for which  $f(\mathbf{x}^L) = \mathbf{u}^L$  is called local scalarization optimizer,
- any  $\mathbf{u}^L$  for which  $B(\mathbf{u}^L) = \mathbb{R}^m$  is called global scalarization optimum.

The requirement of  $\Psi(\mathbf{u}^L) < \Psi(\mathbf{v})$  for all  $\mathbf{v} \in V(\mathbf{u}^L)$  in Definition 59 implies that the local scalarization optimum is strict. This condition eliminates local optima that are located on plateaus. All optimization problems that are considered within this work do not exhibit plateaus, which is why this simplification is made. Nonetheless, the notion of local scalarization optima as of Definition 59 could be extended to also include optima that are located on plateaus. Furthermore, a scalarized MOOP may have more than one point that possesses

the globally optimal scalarization value as shown in Figure 6.1. The global optimum is not unique in such instances and the condition  $\Psi(\mathbf{u}^L) < \Psi(\mathbf{v})$  must be weakened to  $\Psi(\mathbf{u}^L) \leq \Psi(\mathbf{v})$  for allowing multiple global optima to exist as of Definition 59.

### 6.1.2. Basins of Attraction

Basins of attraction are an important concept in MMO that allow the development of efficient search strategies for local scalarization optima. A basin of a local scalarization optimum  $\mathbf{u}^L$  consists of all feasible points  $\mathbf{v} \in \mathcal{Y}$  for which there exists a search path from  $\mathbf{v}$  to  $\mathbf{u}^L$  that is descending in the scalarization space. This means that there exists a curve from  $\mathbf{v}$  to  $\mathbf{u}^L$  in  $\mathcal{Y}$  such that when moving along the curve from  $\mathbf{v}$  to  $\mathbf{u}^L$  the scalarization values of the curve elements decrease (Definition 60 and Figure 6.1). A local search algorithm that starts its search in the basin of  $\mathbf{u}^L$  is expected to converge to  $\mathbf{u}^L$ .

**Definition 60** (basin [BHSS17]). *The basin  $\mathcal{B}(\mathbf{u}^L)$  of a local scalarization optimum  $\mathbf{u}^L$  is a subset of the feasible objective space such that for all  $\mathbf{v} \in \mathcal{B}(\mathbf{u}^L)$  there exists a continuous function  $\alpha : [0, 1] \rightarrow \mathbb{R}^m$  such that  $\alpha(0) = \mathbf{v}$  and  $\alpha(1) = \mathbf{u}^L$  and for all  $t_1, t_2 \in [0, 1]$  with  $t_1 < t_2$  it holds that  $\Psi(\alpha(t_1)) \geq \Psi(\alpha(t_2))$ .*

- The set of all basins for a given objective function of an MOOP and scalarization function is called basin system.

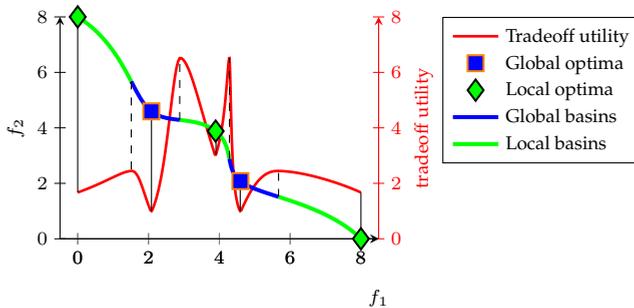


Figure 6.1.: Illustration of local scalarization optima and their corresponding basins. Basins have been restricted to the Pareto front for clarity. Five basins – three local (green) and two global (blue) – are depicted in the figure. Dashed lines mark the boundary of the basins as indicated by the scalarization function (tradeoff utility). The solid lines indicate the positions of the optima of the scalarization function.

### 6.1.3. Local Scalarization Optima Preference Predicate

Local scalarization optima can be formulated as preference predicate by extending the scalarization predicate (Definition 29). The preferred set identified by a scalarization predicate consists of all scalarization optimizers – decision vectors that map to objective vectors that possess the smallest scalarization value among all Pareto optimal points. The preferred set of a local scalarization optima preference predicate additionally contains those decision vectors that map to local scalarization optima (Definition 61). A preference predicate that contains decision vectors that map to local and global scalarization optima is obtained by restricting the  $\leq$ -relation that orders the scalarization space. Given two point  $\mathbf{u}, \mathbf{v} \in \mathcal{Y}$ ,  $\mathbf{u}$  dominates  $\mathbf{v}$  in the scalarization space if  $\mathbf{u}$  possesses a smaller scalarization value than  $\mathbf{v}$  and additionally if  $\mathbf{u}$  and  $\mathbf{v}$  share a common basin. By applying this restriction, the image of any point  $\mathbf{u}$  can only dominate other elements of the scalarization space, whose preimages are located in the same basin as  $\mathbf{u}$ . Elements, whose preimages come from different basins are thereby always non-dominated to each other. Thereby, any strictly local optimum is not dominated by a global optimum as it is the case under the scalarization predicate.

**Definition 61** (local scalarization optima preference predicate [BHSS17]). *Let  $f = (X, Y, F)$  be the objective function of an MOOP and  $\Phi^s = (f^s, \leq)$  with  $(X, \mathbb{R}, F_s)$  be a scalarization predicate. The corresponding local scalarization optima preference predicate is given by  $(f^s, <_L)$ , where for all  $\mathbf{u}, \mathbf{v} \in \mathcal{Y}_p$  the relation  $\Psi(\mathbf{u}) <_L \Psi(\mathbf{v})$  holds iff  $\Psi(\mathbf{u}) \leq \Psi(\mathbf{v})$  and there exists a local scalarization optimum  $\mathbf{u}^L$  such that  $\mathbf{u}, \mathbf{v} \in \mathcal{B}(\mathbf{u}^L)$ .*

## 6.2. Algorithmic Framework for Approximating Scalarization Optima

An algorithmic framework for approximating local scalarization optima is proposed in this section. The framework uses ideas and guidelines that have been established in single objective MMO. Minimizing a scalarized MOOP is closely related to finding the local optima of an SOOP. There exist, however, conceptual differences that need to be addressed when translating single-objective MMO techniques to scalarization in MOO. These difference are discussed in the course of this section. In single objective MMO, the application of a multimodal optimization algorithm is divided into three steps: 1) sampling, 2) basin

## 6.2. Algorithmic Framework for Approximating Scalarization Optima

identification and 3) local optimization [Pre15, Wes15, WRP16].<sup>1</sup>

**Sampling the search space** Single objective multimodal optimization techniques commence by sampling the search space to estimate the topology of the objective function. Local optimizers might be distributed across the entire feasible set or be narrowly located next to global optimizers. In the former case, the sample must cover the entire feasible set. In the latter case, the resolution of the sample must be fined-grained such that the local optimizers can be detected. In both scenarios, the search space must be sufficiently explored such that no optimizer is missed.

**Basin identification** As mentioned in the first step, the samples generated in the previous step serve as input to build an estimate of the topology of the objective function in single-objective MMO and correspondingly the scalarization function in MOO. The estimate is used to identify basins of attraction of local optima. The sample generated in the first step is clustered for this purpose. Each cluster is expected to represent a single basin of attraction and the point in the cluster possessing the smallest scalarization value constitutes a rough estimate of the corresponding local optimum. It may happen, though, that the clustering fails. Either individual basins are missed or clusters cover multiple basins. These issues are discussed in detail in Section 6.3.1.

**Local optimization** Once the clustering is completed, each cluster is used as starting point for a local optimization procedure. If an EA is applied, for example, each cluster serves as start population. If the cluster is completely contained in a single basin, the local optimization algorithm is expected to correctly approximate the corresponding local optimum. Special care must be given to the parametrization of the local search algorithm such that it does not escape its assigned basin and converge to an optimizer possessing a smaller objective value. Local search algorithms for finding local scalarization optima are discussed in Section 6.3.2.

The proposed framework adapts these three steps and modifies them to take special considerations into account that occur in MOO (Algorithm 11). In SOO, local optima may be found across the entire feasible set. In MOO on the other hand, all local scalarization optima must be Pareto optimal and are therefore located on the Pareto front. It therefore makes sense, to focus the sampling on the Pareto optimal set instead of the entire feasible set. Since the Pareto optimal set is not known prior to any optimization effort, it is proposed to compute a

---

<sup>1</sup>The literature on multimodal optimization usually combines sampling and basin identification into a single step, speaking of a two-stage model. Since sampling takes a more prominent role in the light of scalarized preferences, it is introduced as separate step.

## 6. Multimodal Scalarized Preferences

uniform finite set of points approximation of the Pareto front, which serves as input to the basin identification step.

Additionally, scalarization functions that violate Pareto compliance (Definition 44) do not possess a meaningful topology for the entire feasible set that can be exploited for local optimization. The Nash bargaining solution (Definition 39), for example, can only be computed for points that dominate the Nadir point. Scalarization functions that violate binary independence (Definition 45) also require implicit knowledge of the Pareto front, e.g. in the form of reference points. Angle utility (Definition 41), for example, requires knowledge of the extreme points that can be inferred from a uniform finite set of points approximation of the Pareto front. Starting the search from a random sample in the feasible set therefore appears ill-advised, which is why a uniform sample of the Pareto front is considered instead.

Clustering methods applied in single-objective MMO can be adapted in MOO. In SOO, however, local optima are defined in the search space, i.e. a local optimizer must have the smallest objective value in a neighborhood of the feasible set. Local scalarization optima, on the other hand, are defined in the objective space, i.e. a local optimum has to possess the smallest scalarization value among all points in a neighborhood that is contained in the feasible objective space. Therefore, clustering methods need to estimate the topology of the scalarization function with respect to the objective space instead of the search space.

Many basin identification techniques require that the sample is uniformly distributed across the search space to obtain optimal results [WRP16]. Since any non-degenerate Pareto front is an  $m - 1$  manifold in an  $m$ -dimensional vector space, any uniform approximation of the Pareto front constitutes a highly biased sample that is not equally distributed in  $\mathbb{R}^m$ . Whether this circumstance presents itself as limitation to applying single objective basin identification techniques in MOO is the subject of study in Section 6.3.1.

Local optimization algorithms from SOO can be applied by choosing the scalarization function as objective function for optimization. Approximations of reference points required for computing scalarization values can be inferred from the Pareto front approximation in the first step. In case of tradeoff utility (Definition 42) the entire finite set of points approximation can serve as estimate of  $\mathcal{Y}_p$ . If the scalarization function is not Pareto compliant (Proposition 4), the Pareto order must be imposed on top of the comparison of scalarization values. Given  $\mathbf{u}, \mathbf{v} \in \mathcal{Y}$ , the point  $\mathbf{u}$  is only better than  $\mathbf{v}$  if either  $\mathbf{u}$  Pareto dominates  $\mathbf{v}$  or if both objective vectors are non-dominated to each other and  $\mathbf{u}$  possesses a smaller scalarization value than  $\mathbf{v}$ .

### 6.3. Computational Analysis of Framework Components

Table 6.1.: Key differences between MMO in SOO and scalarized MOO.

	Single-objective	Multi-objective
Function values depend on	Decision vector	Objective vector and (potentially) composition of Pareto front
Comparison of candidate solutions	Function value	Pareto domination and scalarization value
Set in which optima/basins are defined	Search space	Objective space
Feasible locations of optima	Feasible set	Pareto front
Samples generated in	Feasible set	Pareto optimal set
Sample composition	Uniform in search space	Biased in objective space
Clustering performed in	Search space	Objective space

The local search must be carefully configured such that the algorithm does not escape its assigned basin. This is a known problem in single-objective MOO [Pre15] and expected to apply to scalarization functions as well. Local optimization is studied in Section 6.3.2. A summary of the differences of MMO in SOO and scalarized MOO is given in Table 6.1.

---

**Algorithm 11:** Local Scalarization Optima Approximation Procedure cf. [BHSS17]

---

**Input :** MOOP  $f$ , scalarization function  $\Psi$

**Output:** Approximation of local scalarization optimizers  $\{\mathbf{x}^{L,i}\}_{i=1}^k$

- 1 Generate uniform Pareto front approximation  $P$
  - 2 Cluster  $P$  into basin estimates  $\mathcal{C} = \{C_1, \dots, C_k\}$
  - 3 **forall**  $i \in [k]$  **do**
  - 4     Run local optimization with  $C_i$  as initial set to approximate  $\mathbf{x}^{L,i}$
  - 5 **return**  $\{\mathbf{x}^{L,1}, \dots, \mathbf{x}^{L,k}\}$
- 

## 6.3. Computational Analysis of Framework Components

The three steps sampling, basin identification and local optimization constitute individual components of an algorithmic framework for approximating local scalarization optima. Different techniques can be applied in each step. The scope of this section is finding suitable methods for each component by conducting a quantitative computational study. For this purpose, techniques that

## 6. Multimodal Scalarized Preferences

have been established in single-objective MMO are adapted for scalarized preferences in MOO. The analysis is performed within individual steps not taking dependencies between methods of different components into account. While there might exist synergies or antagonistic effects between methods of different components, it is more reasonable to assume that any method that performs well in a given step delivers better input to methods in successive steps than techniques that perform worse in the predecessor step. For example, if more basins are detected through clustering local optimization should find more local optima.

The analysis is performed chronologically in the execution order of the steps in Algorithm 11. The method that yields the best performance in the current step is utilized in generating inputs for successive steps in subsequent studies. An extensive study of different sampling techniques is omitted, since the literature on MOO contains a myriad of studies on algorithms seeking uniform finite set of points approximations. Instead, the results of Section 5.3.2 are used for justifying a reasonable approach to sampling that is described in Section 6.3.1. All algorithms are implemented in jMetal and the code is available online [Bra].

### 6.3.1. Basin Identification Methods

Two basin identification methods from single-objective MMO – topographical selection (TS) [TV92] and nearest better clustering (NBC) [Pre15] – are adapted for the use with scalarized preferences in MOO and computationally analyzed. Both techniques have been found to be successful in detecting basins, their performance however greatly depends on the chosen parameters [WRP16]. NBC and TS are tested with different parameter configurations. The best configurations of each algorithm are compared to each other to deliver a final verdict on which method yields the overall best performance.

Based on the results of Section 5.3.2, ESPEA is chosen as method to generate the Pareto front approximations that are used as sample for the basin identification methods. Among all tested algorithms in Section 5.3.2, ESPEA outperformed all other algorithms with the exception of SPEA2, however no significant performance difference between ESPEA and SPEA2 was found. The majority of the test problems considered in the analysis of this chapter have also been part of the study in Section 5.3.2. ESPEA is therefore an adequate technique for computing a sample of the Pareto front. Any other algorithm generating uniform finite set of points approximations, however, could be used instead.

An archive size of 200 was chosen to obtain a sufficiently large cover of the Pareto front. ESPEA was run for 25 000 function evaluations on each problem

### 6.3. Computational Analysis of Framework Components

instance, since the IGD performance indicator improves only marginally in subsequent iterations on the considered test problems. Latin hypercube sampling was used to generate the initial population. The search operators utilized in ESPEA were configured in the same manner as for the study in Section 5.3.2. ESPEA was run 100 times on each problem instance. See Table E.3 for a summary of the ESPEA configuration. Both basin identification algorithms were applied to each final archive of all 100 runs.

#### 6.3.1.1. Clustering Algorithms

Both basin identification techniques that are regarded in the study perform their clustering by building a directed graph from the elements of the Pareto front approximation. For simplicity, the Pareto front approximation is referred to as population in the following. Each population member constitutes a vertex of the graph. The clustering methods connect the vertices by edges according to rules that depend on their scalarization values and their distances in the objective space. The basin identification techniques divide the graph into subgraphs, where the vertices of each subgraph represent a cluster that is an estimate of a basin of attraction.

Recall that  $f(\mathbf{x}^i) := \mathbf{f}^i$ . TS builds the graph in the following way (Algorithm 12). For each population member  $\mathbf{x}^i$  its  $k^c$ -closest neighbors in the objective space according to the Euclidean norm are considered. Among all  $k^c$ -closest neighbors of  $\mathbf{x}^i$ , an edge pointing from  $\mathbf{x}^i$  to the closest of these neighbors that possesses a smaller scalarization value than  $\mathbf{f}^i$  is added to the graph if such a point exists. Each vertex that has an outdegree of zero after all edges have been added is an estimate of a local scalarization optimum. For each such vertex  $v$  a subgraph is identified that consists of all vertices and edges that form part of a path, whose endpoint is  $v$ . These subgraphs are disconnected from each other, since each node possesses at most one outgoing edge. Thereby, each disconnected subgraph represents a cluster.

The description of TS in Algorithm 12 is based on [WRP16] and has already been modified for the purpose of this study. TS was originally designed to identify solely local optima and not their surrounding basins. Therefore, an alteration was introduced to the original algorithm. The original TS clustering algorithm generates incoming edges to all  $k^c$ -closest neighbors possessing a smaller objective (scalarization) value instead of only the smallest as in Algorithm 12 (Line 7). Such a procedure, however, allows individual vertices to form part of multiple clusters, since each vertex may possess up to  $k^c$  outgoing edges. Overlapping clusters would deter the local search, since local optimization algorithms are more prone to escape their assigned basin.

## 6. Multimodal Scalarized Preferences

---

**Algorithm 12:** topographical selection cf. [TV92]

---

**Input** : Pareto front approximation  $P := \{\mathbf{x}^i\}_{i=1}^N$ , scalarization function  $\Psi$ , parameter  $k^c$

**Output:** Clustering  $\mathcal{C}$

- 1 Create directed graph  $G := (V, E)$  with  $V := P$  and  $E := \emptyset$
  - 2 **forall**  $\mathbf{x}^i \in P$  **do**
  - 3      $\mathbf{s} := \text{argsort}(\|\mathbf{f}^1 - \mathbf{f}^i\|_2, \dots, \|\mathbf{f}^N - \mathbf{f}^i\|_2, \leq)$
  - 4      $Q := \{\mathbf{x}^{\mathbf{s}(2)}, \dots, \mathbf{x}^{\mathbf{s}(k^c+1)}\}$  //  $k^c$  closest neighbors
  - 5      $\mathbf{y}^* := \arg \min_{\mathbf{y} \in Q: \Psi(\mathbf{f}^i) > \Psi(f(\mathbf{y}))} (\|\mathbf{f}^i - f(\mathbf{y})\|_2)$
  - 6     **if**  $\mathbf{y}^* \neq \emptyset$  **then**
  - 7          $E := E \cup \{(\mathbf{x}, \mathbf{y}^*)\}$
  - 8 **return**  $\{V' \subseteq V \mid G' = (V', E') \text{ is a disconnected subgraph of } G\}$
- 

Another change involves the distance computation. The original TS considers distances in the search space, whereas in Algorithm 12 distances are measured in the objective space. In the original description of TS, another edge from all of the  $k^c$ -closest neighbors to  $\mathbf{x}^i$  is added to the graph if  $\Psi(\mathbf{f}^i) < \Psi(f(\mathbf{y}))$  and there already exists a path in the graph such that  $\mathbf{x}^i$  can be reached from  $\mathbf{y}$ . This step is intentionally omitted in Algorithm 12, since it does not change the clustering outcome.

The second clustering technique that is considered for the study is NBC (Algorithm 13). NBC builds a spanning tree from the Pareto front approximation. For each population member  $\mathbf{x}$ , an edge to its closest neighbor  $\mathbf{y}^*$  in the objective space that possesses a smaller scalarization value than  $f(\mathbf{x})$  is added to the graph. The distance according to the Euclidean norm between both vectors serves as weight of the edge  $(\mathbf{x}, \mathbf{y}^*)$ . The only element of the Pareto front approximation that does not possess an outgoing edge is the point whose image possesses the smallest scalarization value, which serves as estimate of the global scalarization optimum.

Subsequently, two cutting rules are applied to divide the spanning tree into subgraphs. Cutting rule 1 defines a threshold length  $w^t$ , which is equivalent to the average arc length in the spanning tree multiplied by a parameter  $\phi^c$ . All edges that are larger than the threshold length are cut. The reasoning behind cutting rule 1 is that very long edges are likely to connect local optima to basins of other local optima that possess smaller scalarization values.<sup>2</sup>

---

<sup>2</sup>Preuss [Pre15] further multiplies the threshold length by a correction factor if the sample adheres to complete spatial randomness. Since Pareto front approximations are highly biased samples in  $\mathbb{R}^m$  and do not satisfy complete spatial randomness, the correction factor is dropped.

### 6.3. Computational Analysis of Framework Components

Cutting rule 2 regards all nodes in the graph with three or more incoming edges. For any such node  $\mathbf{x}$ , if the length of its outgoing edge is longer than the median length of its incoming edges multiplied by a parameter  $b^c$ , the outgoing edge is removed from the graph. Cutting rule 2 follows the same logic as cutting rule 1 in identifying nodes that are potentially local optima. The second rule, however, is more likely to identify an approximate local optimum if it is closely located next to another basin that features smaller scalarization values. The edge length of the approximate local optimum to its closest better neighbor would be too small to fall under the threshold length of cutting rule 1. Cutting rule 2 is expected to play a more important role as the number of objectives increases. For two objectives, any node can have at most two incoming edges if the Pareto front approximation consists only of non-dominated points, since the approximation is totally ordered from left to right.

---

**Algorithm 13:** nearest better clustering cf. [BHSS17, Pre15]

---

**Input :** Pareto front approximation  $P$ , scalarization function  $\Psi$ , parameter  $\phi^c$ , parameter  $b^c$

**Output:** Clustering  $\mathcal{C}$

```

1 Create weighted, directed graph  $G := (V, E, w)$  with  $V := P$  and  $E := \emptyset$ 
2 forall  $\mathbf{x} \in P$  do // build spanning tree
3    $\mathbf{y}^* := \arg \min_{\mathbf{y} \in P: \Psi(f(\mathbf{y})) < \Psi(f(\mathbf{x}))} \|f(\mathbf{x}) - f(\mathbf{y})\|_2$ 
4   if  $\mathbf{y}^* \neq \emptyset$  then
5      $E := E \cup \{(\mathbf{x}, \mathbf{y}^*)\}$ 
6      $w((\mathbf{x}, \mathbf{y}^*)) := \|f(\mathbf{x}) - f(\mathbf{y}^*)\|_2$ 
7  $w^t := \phi^c \cdot \sum_{e \in E} w(e) / |E|$  // threshold length
8 forall  $\mathbf{x} \in V : (\deg^-(\mathbf{x}) \geq 3) \wedge (\deg^+(\mathbf{x}) > 0)$  do // cutting rule 2
9   Let  $e^+$  denote the outgoing edge of  $\mathbf{x}$ 
10  Let  $e_1^-, \dots, e_k^-$  denote the incoming edges of  $\mathbf{x}$ 
11  if  $w(e^+) > b^c \cdot \text{median}\{w(e_1^-), \dots, w(e_k^-)\}$  then
12     $E := E \setminus \{e^+\};$ 
13 forall  $e \in E : w(e) > w^t$  do // cutting rule 1
14    $E := E \setminus \{e\}$ 
15 return  $\{V' \subseteq V \mid G' = (V', E')$  is a disconnected subgraph of  $G\}$ 

```

---

Cutting rule 2 is executed before cutting rule 1, since rule 1 might cut incoming edges negating the requirement of having an outgoing edge count of at least three for applying cutting rule 2. The threshold length of cutting rule 1, however, is computed with respect to all edges of the spanning tree, which is why it must be computed before applying cutting rule 2. Historically, cutting rule 2 was introduced after cutting rule 1 in [Pre15], which explains the chosen

## 6. Multimodal Scalarized Preferences

naming scheme.

Both TS and NBC possess parameters that influence the clustering outcome. The neighborhood size  $k^c$  in TS determines the number of basins that can be found, since any vertex with no outgoing edge must have a better scalarization value than all of its  $k^c$  neighbors. Decreasing the neighborhood size increases the likelihood of finding small basins. If the Pareto front approximation is not close enough to the actual front, however, small neighborhood sizes might lead to the identification of multiple clusters in the same basin, since the scalarization landscape may possess multiple small hills and valleys.

The threshold value  $\phi^c$  controls how far clusters need to be located away from each other to qualify as basin estimate. Similar to  $k^c$ , decreasing  $\psi^c$  increases the likelihood of generating more clusters. The effect of  $\phi^c$ , however, is more difficult to understand as arc lengths greatly depend on the quality of the Pareto front approximation. Extensive simulations studies were performed by Preuss in [Pre15] to derive a formula for determining optimal values for  $b^c$  that depends on the sample size  $N = |P|$  and the number of objectives (decision variables in the original publication):

$$b^c(N, m) = (-4.69 \cdot 10^{-4}n^2 + 0.0263m + 3.66m^{-1} - 0.457) \cdot \log_{10}(N) + 7.51 \cdot 10^{-4}n^2 - 0.0421m - 2.26m^{-1} + 1.84. \quad (6.1)$$

Because of the intricate design of Equation (6.1) and the negligible effect that cutting rule 2 is expected to have on NBC in the current scenario, only the variation of  $\phi^c$  is considered in the computational study. The values for parameters  $k^c$  and  $\phi^c$  have been chosen to reflect meaningful choices that have been tested in previous studies [Pre15, WRP16].

### 6.3.1.2. Basin Identification Test Problems

The problems considered in this study were chosen such that they possess multiple local scalarization optima across different scalarization functions. The DEB2DK, DEB3DK and DO2DK problems (Table 5.2) are reasonable choices in this respect, since the parameter  $k$  allows controlling the number of convex bulges their Pareto fronts exhibit. By increasing the number of bulges, the scalarization landscapes undergoes great changes and more scalarization optima are generated (Table 6.2). DTLZ7 and ZDT3 (Table 5.9) are chosen as well, since the Pareto fronts of both problems are disconnected. At least one scalarization optimum is located in each disconnected region.

### 6.3. Computational Analysis of Framework Components

The scalarization functions presented in Section 3.1 have been chosen for the study. As discussed in Section 3.1, these scalarization functions are representative of the different scalarization methodologies available in the literature. The sum of objectives (Definition 33), Nash bargaining solution (Definition 39), angle utility (Definition 41) and tradeoff utility (Definition 42) have been considered in particular. The Chebyshev method was omitted, because it only possesses a single global scalarization optimum on all bi-objective problems of the study. The product of objectives is not considered, since it identifies only boundary points of the given test problems as scalarization optima. An overview of the number of local scalarization optima for the different problems and scalarization functions is given in Table 6.2.

As evidenced by Table 6.2 and Figure 6.2, the chosen scalarization function has a major influence on the shape of the scalarization landscape and the number of scalarization optima. It is therefore imperative, that both the effect of the shape of the Pareto front and the scalarization function is considered in the analysis.

Table 6.2.: Overview of the problems chosen for the multimodal preference study. Identifiers of the scalarization functions are found in Table 3.1. The variable  $m$  denotes the number of objectives.

	$\Psi^s$	$\Psi^n$	$\Psi^a$	$\Psi^t$	$m$
DEB2DK $k = 1$	3	1	1	3	2
DEB2DK $k = 2$	4	2	2	5	2
DEB2DK $k = 3$	5	3	3	7	2
DEB2DK $k = 4$	6	4	4	9	2
DEB3DK $k = 1$	5	1	2	6	3
DEB3DK $k = 2$	10	4	4	23	3
DO2DK $k = 1$ $s = 1$	1	1	1	2	2
DO2DK $k = 2$ $s = 1$	2	1	1	3	2
DO2DK $k = 3$ $s = 1$	2	1	1	5	2
DO2DK $k = 4$ $s = 1$	3	2	1	5	2
DTLZ7	4	4	5	12	3
ZDT3	5	5	5	6	2

#### 6.3.1.3. Basin Identification Performance Indicators

An ideal clustering should group the objective vectors of the Pareto front approximation such that the elements of each cluster are contained in a single basin of attraction. Additionally, each basin is covered by at most one cluster.

## 6. Multimodal Scalarized Preferences

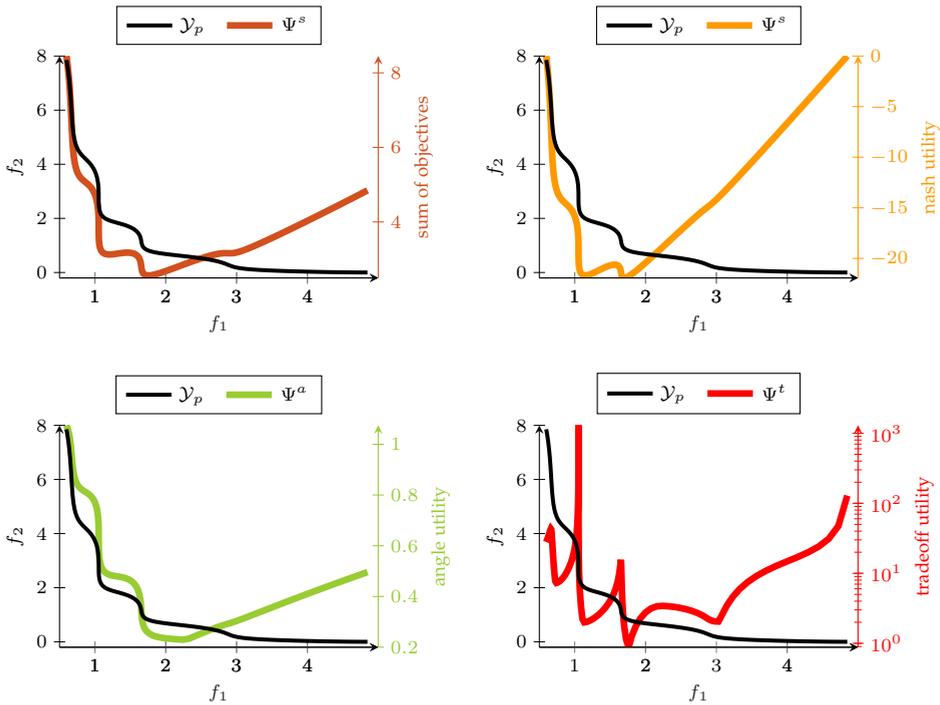


Figure 6.2.: Comparison of scalarization landscapes for the sum of objectives, the Nash bargaining solution, angle utility and tradeoff utility of DO2DK  $k = 4$   $s = 1$ . Individual scalarization functions shape the scalarization landscape in different ways by introducing hills and valleys at different regions of the Pareto front.

If the Pareto front approximation is uniform and consists of sufficient points, all basins should be found through the clustering. In real-world applications, such a clustering is usually difficult to achieve [Pre15]. Preuss [Pre15, Section 4.6.3] distinguishes between three error types to characterize imperfect clusterings:

**Type 1:** At least one basin is not covered by any cluster and remains undetected.

**Type 2:** At least one cluster covers more than one basin.

**Type 3:** Multiple clusters cover a single basin.

Type 1 errors occur if the Pareto front approximation misses entire regions that

### 6.3. Computational Analysis of Framework Components

feature individual basins. In this case, the methodology that is used for approximating the Pareto front does not locate any points in the respective basin. This behavior is the result of either the method failing to generate a uniform Pareto front approximation or if the basin is too small to be identified by the chosen granularity of the approximation. If the chosen population size of an EA is too small, for example, it is possible that even a uniform approximation misses basins that are located between neighboring points (Figure 6.3).

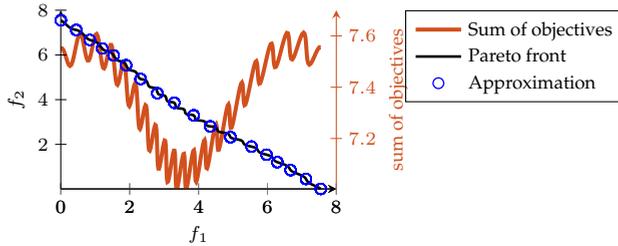


Figure 6.3.: Example of a highly multimodal scalarization landscape that makes basin identification impractical. Although the Pareto front approximation is uniform there exist multiple basins that are not covered by any solution rendering it impossible for any clustering algorithm to detect all basins.

In both cases, type 1 errors are attributed to issues regarding the sampling methodology. The clustering methods cannot remedy the circumstance that no points are located in the respective basins. Any uniform Pareto front approximation of reasonable size should locate sufficient points in all basins. In case a basin is indeed too small for being detected, one might argue from a normative perspective that the corresponding local scalarization optimum is less desirable to the DM. A small basin implies that the corresponding optimum is not robust and it might therefore be difficult to implement in practice [BS07]. If the optimum is strictly local, it is reasonable to assume that other solutions exist close-by that possess a smaller scalarization value [Pre15].

Type 2 errors appear if the clustering is too coarse. The local optimization procedure that is subsequently applied is expected to approximate either of the local optima of the basins covered by the clustering. Depending on the chosen search method, it is likely that of all basins, the local optimum featuring the smallest scalarization value is identified, since local search methods are usually greedy and therefore more attracted to regions featuring smaller scalarization values [Sia16]. If the points of the cluster do not show great variation in their scalarization values between basins, it is reasonable to assume that the local optimum of the basin featuring the most points is approximated, if population-based approaches are employed as local search [Pre15]. In all cases,

## 6. Multimodal Scalarized Preferences

local scalarization optima are likely to be missed, since local search algorithms usually only approximate a single local optimum.

If the clustering is too fine-grained Type 3 errors may occur. Local searches using clusters covering the same basin are expected to converge to the same local optimum. From a decision-making perspective, this does not present itself as a problem, since no information about local scalarization optima is lost. From a computational perspective, however, this situation is highly undesirable, since resources are wasted on approximating the same local optimum multiple times.

In order to compare clusterings generated by different algorithms, it is imperative to define measures for grading the usefulness of a specific clustering. In the same manner in which quality indicators are used to evaluate the uniformity and convergence of a Pareto front approximation, performance indicators related to the different error types have been developed to assess clustering quality. The useful cluster fraction (UCF) and detected basin fraction (DBF) used in single-objective MMO are adopted and modified for this purpose [Pre15].

The following notation is used for the formal definition of the performance indicators:

$\mathcal{B}, B_i$  The variable  $\mathcal{B}$  denotes a basin system as of Definition 60 and  $B_i$  is the identifier for a single basin of the basin system.

$\mathcal{C}, C_j$  The set of clusters – the clustering – generated by a clustering algorithm is denoted by  $\mathcal{C}$  and the  $C_i$  are individual clusters of  $\mathcal{C}$ . The clustering is a partitioning of a finite set of points approximation of the Pareto front. Each cluster is thereby a finite subset of the objective space  $\mathcal{Y}$ .

Both UCF and DBF rely on the concept of decided clusters. A cluster is called decided if more than half of its elements are located in a single basin of attraction (Definition 62). Said basin is called the main basin of the cluster. The underlying assumption of the UCF and DBF is that given a specific cluster, a local search converges towards the local optimum of the main basin. If there exists no main basin and a metaheuristic is applied as local search technique, it is more likely that local optima of different basins are identified if the search is repeated using the same cluster as initial set. The concept of decided clusters is therefore used to penalize severe cases of type 2 errors in the performance evaluation.

**Definition 62** (decided cluster [Pre15]). *A cluster  $C_j$  is called decided if there exists a basin  $B_i \in \mathcal{B}$  and a subset  $D \subseteq C_j$  such that  $D \subseteq B_i$  and  $|D| > |C_j|/2$ . Furthermore,  $B_i$  is called the main basin of  $C_j$ .*

### 6.3. Computational Analysis of Framework Components

Since multiple decided clusters may cover the same basin (type 3 error), the notion of useful clusters is considered next. If multiple decided clusters cover the same basin, one can argue that only one of them is useful, since local search is expected to converge to the same local optimum if either cluster is used as initial set of the search. The number of useful clusters therefore consists of the number of basins that are covered by at least one decided clusters (Definition 63).

**Definition 63** (number of useful clusters [Pre15]). *The number of useful clusters is given by*

$$N^c = |\{B_i \in \mathcal{B} \mid \exists C_j \in \mathcal{C} : B_i \text{ is the main basin of } C_j\}|. \quad (6.2)$$

The UCF is then defined as the number of useful clusters divided by the total number of clusters (Definition 64). A larger UCF implies that more clusters of the clustering are useful. An increase in the number of clusters may result in obtaining more clusters that feature common main basins. This is especially the case if the number of clusters exceeds the number of basins. Consequently, obtaining more clusters elicits a higher computational effort, since local optimization is performed for each cluster. The UCF therefore penalizes type 3 errors.

**Definition 64** (useful cluster fraction [Pre15]). *The useful cluster fraction (UCF) is defined as*

$$UCF(\mathcal{C}) = \frac{N^c}{|\mathcal{C}|}. \quad (6.3)$$

The DBF is computed by dividing the number of useful clusters by the total number of basins. A DBF of 1 implies that each basin is covered by at least one decided cluster. If a decided cluster is obtained for each basin, local optimization is expected to find all local optima. The DBF is indifferent towards multiple clusters sharing the same basin. The more clusters are identified, the likelier it is that for any basin  $B_i$  there exists a cluster  $C_j$  such that  $B_i$  is the main basin of  $C_j$ . In this sense, the DBF penalizes type 2 errors.

**Definition 65** (detected basin fraction cf. [Pre15]). *The detected basin fraction (DBF) is defined as*

$$DBF(\mathcal{C}) = \frac{N^c}{|\mathcal{B}|}. \quad (6.4)$$

Note that the definition of the DBF in this work differs from the one proposed in [Pre15]. Preuss divides the number of useful clusters by the number of basins in which the sampling method has generated at least one point. This definition, however, penalizes the sampling method. For example, a clustering that

## 6. Multimodal Scalarized Preferences

consists of one cluster based on a sample that covers only a single basin, has a DBF of 1. If another sample of the same Pareto front covers four basins and three decided clusters are found, the clustering possesses a DBF of 0.75. The latter clustering, however, is undoubtedly more preferable to the former. Simon Wessing has therefore suggested in a personal communication to divide the number of useful clusters by the total number of basins instead.

A perfect clustering – a single decided cluster for each basin - achieves a score of 1 for both the UCF and DBF. At the same time, however, either indicator may be improved at the expense of the other. NBC and TS feature parameters that control the granularity of the clustering. A coarse clustering creates few, a fine-grained clustering more clusters. Coarse clusterings are more likely to achieve a higher UCF and a smaller DBF, whereas fine-grained clusterings will obtain a smaller UCF and a higher DBF. In this sense, a tradeoff can be observed between both performance indicators.

Both indicators may also be interpreted from a computational and decision-making perspective. Similar to the discussion about the error types, the UCF is more important when considering computational costs. Fewer clusters imply less computational effort in the successive local optimization step. The DBF is of greater importance with respect to decision-making. Having more options at her disposal is more desirable for the DM.

Both the UCF and DBF require the knowledge of the basin structure of the problems that are solved. Since not all considered scalarization functions are Pareto compliant (Definition 44), the basins are restricted to the Pareto front. Each basin is discretized by a uniform finite set of points in the same manner in which the Pareto fronts for the computational studies Section 5.3 were generated. An objective vector  $\mathbf{u}$  of a Pareto front approximation is associated with a basin in the following way. The distance between  $\mathbf{u}$  and each point of the discretized basins is computed. The basin of the point to which  $\mathbf{u}$  is closest is then associated with  $\mathbf{u}$ .

For two objectives, the maxima of the scalarization landscape mark the boundary points of the basins (Figure 6.1). In three and higher dimensions, it is possible that there exist Pareto optimal points for which descending paths in the scalarization space can be constructed to multiple local optima. In these cases, each point was assigned such that the path length in the objective space to the local optimum is shortest. Note, that the number of reported minima in [BHSS17, Table 1] and Table 6.2 differ for some of the three objective problems. A different methodology was used to compute the basins and the scalarization

minima in this work that is reported in Appendix D.<sup>3</sup>

#### 6.3.1.4. Discussion of the Basin Identification Study Results

Five different parametrizations have been chosen for TS and NBC;  $k^c \in \{2, 4, 6, 10, 20\}$  and  $\phi^c \in \{1.5, 2, 2.5, 3, 3.5\}$  (Table E.4). The selected values reflect reasonable ranges for both algorithms that have been studied in the past [Pre15, WRP16]. Tables 6.3 to 6.6 show the median and IQR performance of TS and NBC. The figures confirm that there exists a noticeable tradeoff between the UCF and DBF as predicted. Increasing the neighborhood size  $k^c$  or the threshold  $\phi^c$  improves the UCF while the DBF deteriorates. Vice versa, decreasing the neighborhood size or the threshold yields a larger DBF and a smaller UCF. This observation is consistent across all four scalarization functions.

Both parameters  $k^c$  and  $\phi^c$  tend to have a stronger effect on the UCF than on the DBF. Small neighborhoods and thresholds apparently result in the generation of many clusters that share the same main basin. This leads to a large deterioration of the UCF as  $k^c$  and  $\phi^c$  decrease. The performance drops are worst for the tradeoff utility problems, most probably since their scalarization landscapes feature many steep hills and valleys. NBC also appears to be more affected than TS.

On the other hand, increasing the neighborhood size or the threshold has a far smaller effect on the DBF. However, if the values for both parameters would be further increased beyond reasonable values, a similar performance deterioration as for UCF should occur as well. Increasing  $k^c$  or  $\phi^c$  limits the overall number of clusters that can be found. This is mostly evident for TS, where the number of feasible clusters is the size of the Pareto front approximation divided by the neighborhood size. Performance regarding the DBF appears to be very robust with respect to the parametrization as the reported values in Tables 6.4 and 6.6 mostly coincide across the different problems and scalarization functions. The DBF is large on nearly all problem instances and both clustering methods achieve a perfect score of 1 on the majority of all problems.

This allows the conclusion that both TS and NBC are successful in detecting basins. The recommendation for a given clustering method and parametrization depends on the preferences regarding the tradeoff of computational resources and the number of minima found. From a decision-making perspective, the search for local optima should be prioritized to saving computational

---

<sup>3</sup>Preliminary studies revealed that ESPEA occasionally does not approximate all four patches of the Pareto front of DTLZ7. Only runs in which ESPEA finds all patches are considered in the study such that the performance analysis of the clustering algorithms is not deterred by type 1 errors.

## 6. Multimodal Scalarized Preferences

resources. A few local searches finding the same local optimum should be deemed acceptable if more interesting solutions can be generated. At the same time, the additional runs spent on repeatedly identifying the same local optimum should not be disproportionate to the total number of optima found. For example, performing ten local searches for finding two local optima could be deemed reasonable. However, spending additional 90 searches to identify one more local minimum could be considered excessive.

Small neighborhoods and thresholds lead to the detection of nearly all basins, however they also result in the generation of many useless clusters. The UCF drops below 10% on some of the considered test problems. At the same time, increasing either parameter results only in small deteriorations of the DBF. Both  $k^c = 6$  and  $\phi^c = 2.5$  are reasonable compromises among all tested parametrizations. If both configurations are directly compared to each other across all test instances, TS is marginally better than NBC with respect to the DBF. Therefore, the clusters identified by TS with  $k^c = 6$  are used as input for the local optimization in the next study.

Note that no statistical analysis is performed to investigate the performance differences for significance. Median performance across different parametrization coincides very often and show little variation as evidenced by the reported IQRs. This implies that there exists little to no uncertainty in the results that could be attributed to chance. The procedure applied in Section 5.3.2 would report no significant difference between the parametrizations  $k^c = 6$  and  $\phi^c = 2.5$ , since the results are too similar.

Table 6.3.: Basin identification study – UCF of TS. Median and IQR (as subscript) results for different parametrizations of  $k^c$  across 100 runs. Best performances are colored in green, second-best performances in blue. Increasing  $k^c$  steadily improves the UCF, since less clusters are created that are located in the same basin. Performance deteriorates if  $k^c$  is further increased once clusters are merged that are located in distinct basins.

	Sum of objectives					Nash				
	2	4	6	10	20	2	4	6	10	20
DEB2DK $k = 1$	0.33 <sub>0.12</sub>	0.75 <sub>0.15</sub>	1.00 <sub>0.25</sub>	1.00 <sub>0.00</sub>						
DEB2DK $k = 2$	0.50 <sub>0.13</sub>	0.80 <sub>0.20</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.33</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>
DEB2DK $k = 3$	0.63 <sub>0.16</sub>	1.00 <sub>0.17</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.25</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>
DEB2DK $k = 4$	0.67 <sub>0.13</sub>	1.00 <sub>0.14</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	0.80 <sub>0.13</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>
DEB3DK $k = 1$	0.21 <sub>0.05</sub>	0.48 <sub>0.14</sub>	0.80 <sub>0.17</sub>	1.00 <sub>0.17</sub>	1.00 <sub>0.00</sub>	0.07 <sub>0.02</sub>	0.25 <sub>0.13</sub>	0.50 <sub>0.50</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>
DEB3DK $k = 2$	0.33 <sub>0.06</sub>	0.57 <sub>0.13</sub>	0.67 <sub>0.14</sub>	0.70 <sub>0.13</sub>	0.50 <sub>0.17</sub>	0.20 <sub>0.03</sub>	0.57 <sub>0.17</sub>	0.67 <sub>0.33</sub>	0.67 <sub>0.00</sub>	0.67 <sub>0.00</sub>
DO2DK $k = 1 s = 1$	0.50 <sub>0.00</sub>									
DO2DK $k = 2 s = 1$	0.67 <sub>0.17</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.33</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>
DO2DK $k = 3 s = 1$	0.67 <sub>0.33</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	0.50 <sub>0.50</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>
DO2DK $k = 4 s = 1$	0.75 <sub>0.25</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	0.00 <sub>1.00</sub>	0.67 <sub>0.33</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>
DTLZ7	0.17 <sub>0.03</sub>	0.44 <sub>0.15</sub>	0.80 <sub>0.13</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	0.11 <sub>0.02</sub>	0.21 <sub>0.03</sub>	0.31 <sub>0.05</sub>	0.57 <sub>0.17</sub>	1.00 <sub>0.00</sub>
ZDT3	1.00 <sub>0.00</sub>	1.00 <sub>0.17</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>				
	2	4	6	10	20	2	4	6	10	20
DEB2DK $k = 1$	0.17 <sub>0.06</sub>	0.50 <sub>0.17</sub>	1.00 <sub>0.50</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	0.21 <sub>0.04</sub>	0.50 <sub>0.15</sub>	0.75 <sub>0.40</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>
DEB2DK $k = 2$	0.33 <sub>0.11</sub>	0.67 <sub>0.33</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	0.21 <sub>0.04</sub>	0.45 <sub>0.08</sub>	0.71 <sub>0.21</sub>	1.00 <sub>0.17</sub>	1.00 <sub>0.00</sub>
DEB2DK $k = 3$	0.50 <sub>0.17</sub>	1.00 <sub>0.25</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	0.27 <sub>0.05</sub>	0.58 <sub>0.10</sub>	0.78 <sub>0.18</sub>	1.00 <sub>0.09</sub>	1.00 <sub>0.00</sub>
DEB2DK $k = 4$	0.57 <sub>0.17</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	0.33 <sub>0.05</sub>	0.64 <sub>0.13</sub>	0.90 <sub>0.08</sub>	1.00 <sub>0.00</sub>	0.86 <sub>0.16</sub>
DEB3DK $k = 1$	0.04 <sub>0.01</sub>	0.11 <sub>0.04</sub>	0.20 <sub>0.08</sub>	0.33 <sub>0.08</sub>	0.50 <sub>0.50</sub>	0.19 <sub>0.03</sub>	0.32 <sub>0.07</sub>	0.50 <sub>0.11</sub>	0.71 <sub>0.17</sub>	1.00 <sub>0.20</sub>
DEB3DK $k = 2$	0.15 <sub>0.03</sub>	0.40 <sub>0.11</sub>	0.60 <sub>0.23</sub>	0.75 <sub>0.33</sub>	1.00 <sub>0.00</sub>	0.40 <sub>0.07</sub>	0.50 <sub>0.08</sub>	0.55 <sub>0.10</sub>	0.54 <sub>0.16</sub>	0.33 <sub>0.20</sub>
DO2DK $k = 1 s = 1$	0.50 <sub>0.00</sub>	0.07 <sub>0.01</sub>	0.13 <sub>0.03</sub>	0.20 <sub>0.04</sub>	0.50 <sub>0.27</sub>	1.00 <sub>0.00</sub>				
DO2DK $k = 2 s = 1$	1.00 <sub>0.00</sub>	0.10 <sub>0.01</sub>	0.21 <sub>0.05</sub>	0.38 <sub>0.17</sub>	0.88 <sub>0.25</sub>	1.00 <sub>0.00</sub>				
DO2DK $k = 3 s = 1$	0.50 <sub>0.50</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	0.17 <sub>0.02</sub>	0.36 <sub>0.05</sub>	0.63 <sub>0.16</sub>	1.00 <sub>0.17</sub>	1.00 <sub>0.00</sub>
DO2DK $k = 4 s = 1$	0.50 <sub>0.38</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	0.19 <sub>0.03</sub>	0.42 <sub>0.10</sub>	0.71 <sub>0.21</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>
DTLZ7	0.13 <sub>0.02</sub>	0.22 <sub>0.05</sub>	0.29 <sub>0.05</sub>	0.39 <sub>0.08</sub>	0.67 <sub>0.14</sub>	0.28 <sub>0.04</sub>	0.41 <sub>0.07</sub>	0.53 <sub>0.09</sub>	0.67 <sub>0.14</sub>	0.67 <sub>0.26</sub>
ZDT3	0.50 <sub>0.10</sub>	0.83 <sub>0.29</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	0.11 <sub>0.01</sub>	0.19 <sub>0.02</sub>	0.31 <sub>0.06</sub>	0.63 <sub>0.16</sub>	1.00 <sub>0.00</sub>



Table 6.5.: Basin identification study – UCF of NBC. Median and IQR (as subscript) results for different parametrizations of  $\phi^c$  across 100 runs. Best performances are colored in green, second-best performances in blue. The UCF improves if  $\phi^c$  is increases, since less clusters are created that are located in the same basin. Performance deteriorates if  $k^c$  is further increased once clusters are merged that are located in distinct basins.

	Sum of objectives					Nash				
	1.5	2.0	2.5	3.0	3.5	1.5	2.0	2.5	3.0	3.5
DEB2DK $k = 1$	0.38 <sub>0.10</sub>	0.60 <sub>0.25</sub>	0.75 <sub>0.25</sub>	1.00 <sub>0.25</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.50</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>
DEB2DK $k = 2$	0.67 <sub>0.13</sub>	1.00 <sub>0.20</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	0.67 <sub>0.33</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>
DEB2DK $k = 3$	0.83 <sub>0.17</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	0.60 <sub>0.15</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>
DEB2DK $k = 4$	1.00 <sub>0.14</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	0.67 <sub>0.10</sub>	1.00 <sub>0.20</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>
DEB3DK $k = 1$	0.28 <sub>0.06</sub>	0.80 <sub>0.12</sub>	1.00 <sub>0.20</sub>	1.00 <sub>0.17</sub>	1.00 <sub>0.17</sub>	0.04 <sub>0.01</sub>	0.20 <sub>0.08</sub>	0.50 <sub>0.50</sub>	1.00 <sub>0.50</sub>	1.00 <sub>0.00</sub>
DEB3DK $k = 2$	0.44 <sub>0.06</sub>	0.67 <sub>0.18</sub>	0.67 <sub>0.18</sub>	0.67 <sub>0.21</sub>	0.63 <sub>0.21</sub>	0.17 <sub>0.04</sub>	0.80 <sub>0.33</sub>	0.80 <sub>0.33</sub>	0.67 <sub>0.00</sub>	0.67 <sub>0.00</sub>
DO2DK $k = 1 s = 1$	0.10 <sub>0.03</sub>	0.33 <sub>0.25</sub>	0.50 <sub>0.17</sub>	0.50 <sub>0.00</sub>	0.50 <sub>0.00</sub>	0.10 <sub>0.03</sub>	0.33 <sub>0.25</sub>	0.50 <sub>0.17</sub>	0.50 <sub>0.00</sub>	0.50 <sub>0.00</sub>
DO2DK $k = 2 s = 1$	0.11 <sub>0.02</sub>	0.50 <sub>0.27</sub>	0.67 <sub>0.33</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	0.20 <sub>0.07</sub>	0.67 <sub>0.50</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>
DO2DK $k = 3 s = 1$	0.14 <sub>0.04</sub>	0.40 <sub>0.10</sub>	0.67 <sub>0.17</sub>	0.67 <sub>0.00</sub>	0.67 <sub>0.33</sub>	0.05 <sub>0.01</sub>	0.25 <sub>0.13</sub>	0.50 <sub>0.17</sub>	0.50 <sub>0.00</sub>	0.50 <sub>0.38</sub>
DO2DK $k = 4 s = 1$	0.19 <sub>0.04</sub>	0.50 <sub>0.17</sub>	0.67 <sub>0.15</sub>	0.75 <sub>0.08</sub>	0.67 <sub>0.08</sub>	0.17 <sub>0.05</sub>	0.50 <sub>0.10</sub>	0.67 <sub>0.17</sub>	0.67 <sub>0.00</sub>	0.67 <sub>0.33</sub>
DTLZ7	0.24 <sub>0.06</sub>	0.80 <sub>0.20</sub>	1.00 <sub>0.20</sub>	1.00 <sub>0.20</sub>	1.00 <sub>0.20</sub>	0.17 <sub>0.02</sub>	0.33 <sub>0.06</sub>	0.50 <sub>0.13</sub>	0.67 <sub>0.23</sub>	0.80 <sub>0.13</sub>
ZDT3	1.00 <sub>0.00</sub>									
	Angle utility					Tradeoff utility				
	1.5	2.0	2.5	3.0	3.5	1.5	2.0	2.5	3.0	3.5
DEB2DK $k = 1$	0.11 <sub>0.02</sub>	0.25 <sub>0.08</sub>	0.50 <sub>0.17</sub>	1.00 <sub>0.50</sub>	1.00 <sub>0.50</sub>	0.43 <sub>0.13</sub>	0.60 <sub>0.15</sub>	1.00 <sub>0.25</sub>	1.00 <sub>0.25</sub>	1.00 <sub>0.00</sub>
DEB2DK $k = 2$	0.40 <sub>0.17</sub>	1.00 <sub>0.33</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	0.56 <sub>0.13</sub>	0.71 <sub>0.12</sub>	0.83 <sub>0.17</sub>	0.83 <sub>0.17</sub>	1.00 <sub>0.17</sub>
DEB2DK $k = 3$	0.75 <sub>0.15</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	0.70 <sub>0.12</sub>	0.88 <sub>0.10</sub>	0.88 <sub>0.13</sub>	1.00 <sub>0.13</sub>	1.00 <sub>0.13</sub>
DEB2DK $k = 4$	0.80 <sub>0.20</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	0.75 <sub>0.13</sub>	0.90 <sub>0.18</sub>	1.00 <sub>0.10</sub>	1.00 <sub>0.10</sub>	1.00 <sub>0.10</sub>
DEB3DK $k = 1$	0.04 <sub>0.01</sub>	0.17 <sub>0.06</sub>	0.33 <sub>0.23</sub>	0.50 <sub>0.17</sub>	0.50 <sub>0.17</sub>	0.31 <sub>0.06</sub>	0.61 <sub>0.21</sub>	0.80 <sub>0.17</sub>	0.80 <sub>0.15</sub>	0.82 <sub>0.28</sub>
DEB3DK $k = 2$	0.20 <sub>0.03</sub>	0.67 <sub>0.23</sub>	0.75 <sub>0.31</sub>	1.00 <sub>0.25</sub>	1.00 <sub>0.25</sub>	0.57 <sub>0.10</sub>	0.54 <sub>0.15</sub>	0.50 <sub>0.16</sub>	0.45 <sub>0.18</sub>	0.44 <sub>0.16</sub>
DO2DK $k = 1 s = 1$	0.10 <sub>0.03</sub>	0.33 <sub>0.25</sub>	0.50 <sub>0.17</sub>	0.50 <sub>0.00</sub>	0.50 <sub>0.00</sub>	0.17 <sub>0.05</sub>	0.29 <sub>0.08</sub>	0.40 <sub>0.21</sub>	0.40 <sub>0.17</sub>	0.40 <sub>0.15</sub>
DO2DK $k = 2 s = 1$	0.05 <sub>0.01</sub>	0.25 <sub>0.13</sub>	0.50 <sub>0.50</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	0.21 <sub>0.05</sub>	0.38 <sub>0.10</sub>	0.50 <sub>0.17</sub>	0.60 <sub>0.25</sub>	0.75 <sub>0.15</sub>
DO2DK $k = 3 s = 1$	0.05 <sub>0.01</sub>	0.25 <sub>0.13</sub>	0.50 <sub>0.17</sub>	0.50 <sub>0.00</sub>	0.50 <sub>0.38</sub>	0.36 <sub>0.05</sub>	0.53 <sub>0.09</sub>	0.67 <sub>0.13</sub>	0.71 <sub>0.13</sub>	0.80 <sub>0.13</sub>
DO2DK $k = 4 s = 1$	0.06 <sub>0.02</sub>	0.25 <sub>0.13</sub>	0.33 <sub>0.17</sub>	0.50 <sub>0.17</sub>	0.50 <sub>0.00</sub>	0.38 <sub>0.06</sub>	0.56 <sub>0.13</sub>	0.71 <sub>0.21</sub>	0.83 <sub>0.12</sub>	0.83 <sub>0.17</sub>
DTLZ7	0.20 <sub>0.05</sub>	0.31 <sub>0.07</sub>	0.38 <sub>0.11</sub>	0.44 <sub>0.10</sub>	0.50 <sub>0.13</sub>	0.42 <sub>0.06</sub>	0.56 <sub>0.10</sub>	0.62 <sub>0.13</sub>	0.61 <sub>0.14</sub>	0.63 <sub>0.21</sub>
ZDT3	0.71 <sub>0.21</sub>	1.00 <sub>0.17</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	0.24 <sub>0.04</sub>	0.36 <sub>0.05</sub>	0.45 <sub>0.12</sub>	0.50 <sub>0.10</sub>	0.50 <sub>0.17</sub>



## 6.3.2. Local Optimization Algorithms

The modification of four single-objective local optimization algorithms for approximating local scalarization optima is proposed in this section. The greatest challenge during local optimization is to force the algorithm to stay in its assigned basin and not converge to a stronger optimum, i.e. an optimum that possesses a smaller scalarization value [BHSS17]. For this reason, different parameter configurations are tested to find optimal parametrizations of each algorithm. The best configurations of every algorithm are compared to each other to assess which method delivers the best results in the proposed framework. The clusters generated by TS using the neighborhood size  $k^c = 6$  as reported in Section 6.3.1 are used as input for the local optimization algorithms. A population size of 20 was used if applicable. If a cluster possessed more than 20 elements, the 20 members possessing the smallest scalarization value were retained. If a cluster possessed less than 20 elements, it was filled with random copies of the existing cluster members. Each algorithm was run for a maximum of 5000 function evaluation on each cluster.

### 6.3.2.1. Algorithms for Approximating Local Optima

The four algorithms have been chosen to represent different established search techniques that are frequently and successfully applied in stochastic SOO. The Covariance Matrix Adaption Evolutionary Strategy (CMA-ES) developed by Hansen [Han06] is considered to be one of the top performing EAs in BBO and is also frequently used as local search method in single-objective MMO [Pre15]. Recall that  $n$  denotes the number of decision variables. The algorithm utilizes an  $n$ -dimensional vector and an  $n \times n$ -matrix that serve as mean and covariance matrix of a multivariate normal distribution from which decision vectors are sampled. Mean and covariance matrix are updated in each iteration such that the multivariate normal distribution is biased towards the decision vectors of those population members that possess the smallest objective values. An outline of CMA-ES including the proposed modifications is shown in Algorithm 14.

The original CMA-ES initializes the mean by uniform random numbers and the covariance matrix by the identity matrix. This corresponds to a random search in the first iteration. In order to take the information into account regarding the basin structure generated through the clustering, Algorithm 14 utilizes the decision vector of the cluster member possessing the smallest scalarization value as initial distribution mean. The identity matrix is retained as initial covariance matrix to prevent premature convergence. Early tests have revealed that using the covariance matrix of the cluster as initial covariance matrix results

## 6. Multimodal Scalarized Preferences

---

### Algorithm 14: CMA-ES cf. [Han06]

---

**Input** : Cluster  $C = \{\mathbf{x}^i\}_{i=1}^N$ , MOOP  $f$ , scalarization function  $\Psi$ , weights  $\lambda$ , parameters  $\sigma, \alpha, c_c, c_s, c_1, c_\mu, d_\sigma, \mu$

**Output**: Local scalarization optimizer approximation  $\mathbf{y}^b$

- 1  $\mathbf{y} := \arg \min \Psi(\mathbf{f}^i)$  // distribution mean
- 2  $\mathbf{V} := \mathbf{I}_n$  // covariance/identity matrix
- 3  $\mathbf{y}^b := \mathbf{y}$  // best candidate solution
- 4 **repeat**
- 5     Let  $P := \{\mathbf{x}^j\}_{j=1}^N$  with  $\mathbf{x}^j := \mathcal{N}_n(\mathbf{y}, \sigma^2 \mathbf{V})$
- 6      $\mathbf{y}^b := \arg \min_{\{\mathbf{z} \in P \cup \{\mathbf{y}^b\} \mid \forall \mathbf{y} \in P: f(\mathbf{y}) \not\prec_p f(\mathbf{z})\}} \Psi(f(\mathbf{z}))$  // retain best
- 7      $P := \text{sort}(P, (\prec_p, \Psi))$
- 8      $\mathbf{y}^{old} = \mathbf{y}$
- 9      $\mathbf{y} := \text{updateMean}(P, \lambda, \mu)$
- 10     $\mathbf{p}_\sigma := \text{updateIsotropicEvolutionPath}(\mathbf{p}_\sigma, c_\sigma, \mu_w, \mathbf{V}, \mathbf{y}^{old}, \mathbf{y}, \sigma)$
- 11     $\mathbf{p}_c :=$   
      $\text{updateAnisotropicEvolutionPath}(\mathbf{p}_c, c_c, \alpha, \mu_w, \mathbf{p}_\sigma, \mathbf{y}^{old}, \mathbf{y}, \sigma)$
- 12     $\mathbf{V} := \text{updateCovarianceMatrix}(\mathbf{V}, c_1, c_\mu, c_s, \mathbf{p}_c, \lambda, P, \mathbf{y}^{old}, \sigma)$
- 13     $\sigma := \text{updateStepSize}(\sigma, c_\sigma, d_\sigma, \mathbf{p}_\sigma)$
- 14 **until** *stopping criterion*
- 15 **return**  $\mathbf{y}^b$

---

in numerical instabilities of CMA-ES resulting in an early termination of the algorithm.

Let  $N$  once more denote the population size. At the beginning of each iteration,  $N$  individuals are generated by sampling decision vectors from the  $n$ -variate normal distribution  $\mathcal{N}_n(\mathbf{y}, \sigma^2 \mathbf{V})$  (Line 5), where  $\sigma$  is a search strategy parameter called step size, whose meaning is discussed in the subsequent paragraphs. These candidate solutions are sorted according to their scalarization values in ascending order and stored in the population  $P$ . If the scalarization function  $\Psi$  is not Pareto compliant (Definition 44) then a Pareto dominance comparison is performed before scalarization values are compared.

Lines 9 to 13 summarize the update strategy of  $\mathbf{y}$ ,  $\mathbf{V}$  and  $\sigma$  that was retained unchanged from the original formulation of CMA-ES (see Algorithms 20 to 24 in the appendix for details). The update strategy encompasses a series of intricate computations that are independent of the context of local scalarization optima and therefore only discussed briefly. Refer to [Han06, Sia16] for a detailed explanation of the update procedure and the meaning of the different parameters.

### 6.3. Computational Analysis of Framework Components

Summarizing, the CMA-ES described in Algorithm 14 is modified in two ways but otherwise equivalent to the procedure detailed in [Han06]. The first modification consists of selecting the population member having the smallest scalarization value as distribution mean instead of the zero vector. The second modification concerns the selection of the new mean and the sorting procedure in Lines 6 and 7. The modified selection criterion consists of a Pareto domination check and subsequent comparison of scalarization values if both candidate solutions are nondominated to each other. The original procedure in [Han06] only compares objective values, i.e. scalarization values in the current context.

The new mean (Line 9) is generated by computing a weighted sum of the decision vectors of the  $\mu$  best population members, where  $\mu$  is usually chosen as less than half of the population size. CMA-ES uses information contained in the evolution path of the population for updating the covariance matrix and the step size in each iteration. The evolution path records the trajectory of the population in the decision and objective space across all generations. This information is used to update the covariance matrix to bias the normal distribution towards promising regions in the decision space for speeding up the convergence of the algorithm. The step size is controlled using the movement of the distribution mean  $\mathbf{y}$  to prevent premature convergence. Information with respect to the evolution path of the covariance matrix and the step size are stored in the vectors  $\mathbf{p}_\sigma, \mathbf{p}_c \in \mathbb{R}^m$ , respectively. The vector  $\mathbf{p}_\sigma$  is called isotropic evolution path, since the step size  $\sigma$  does not favor movements in particular directions and  $\mathbf{p}_c$  is called anisotropic evolution path, since the covariance matrix controls movements towards favorable regions. Both vectors  $\mathbf{p}_\sigma$  and  $\mathbf{p}_c$  are initialized by the zero vector.

The values for the strategy parameters have been chosen as suggested in the literature and are listed in Table E.6. The step size  $\sigma$  is usually the only parameter that is explicitly set by the user and adapted to the problem that is solved. Smaller step sizes focus the search locally, while larger step sizes broaden the search. As a general rule of thumb, the step size is chosen as a fraction of the bounding box of the decision space [cma].

A generational elitist genetic algorithm (GA) was included in the study. A GA is a specific type of EA that encodes the decision vectors of candidate solutions for the optimization process. The representation of the decision vectors must be decoded before objective values are calculated. The recombination and mutation operators of the GA operate on the encoded candidate solutions instead of the decision vectors. The terms GA and EA, however, are often used synonymously [Sia16, ES<sup>+</sup>03].

An outline of the GA used in this study is presented in Algorithm 15. The GA uses the genetic operators binary tournament selection, SBX and polynomial mutation (see Section 5.3.1). Binary tournament selection chooses two parents

## 6. Multimodal Scalarized Preferences

from the current population using scalarization values and Pareto domination. The parents are recombined by applying SBX and the resulting offspring is transformed using polynomial mutation. The two best candidate solutions in terms of scalarization value and Pareto domination are retained for the successive iteration. A crossover probability of 1.0 and a mutation probability of  $1/n$  was used. As explained in the previous chapter, these values have empirically shown to generate good results.

---

**Algorithm 15:** genetic algorithm (GA) cf. [Sia16, ES<sup>+</sup>03]

---

**Input :** Cluster  $C = \{\mathbf{x}^i\}_{i=1}^N$ , MOOP  $f$ , scalarization function  $\Psi$ , parameter  $\eta$

**Output:** Local scalarization optimizer approximation  $\mathbf{y}^b$

```

1  $P := C$ 
2 repeat
3    $P := \text{sort}(P, (<_p, \Psi))$ 
4    $Q := \{\mathbf{x}^1, \mathbf{x}^2\}$  // Retain best candidate solutions
5   forall  $i \in [N]$  do
6      $(\mathbf{y}^1, \mathbf{y}^2) := \text{binaryTournamentSelection}(P)$ 
7      $\mathbf{z} := \text{simulatedBinaryCrossover}(\mathbf{y}^1, \mathbf{y}^2, \eta)$ 
8      $\mathbf{z} := \text{polynomialMutation}(\mathbf{z}, \eta)$ 
9      $Q := Q \cup \{\mathbf{z}\}$ 
10   $P := Q$ 
11 until stopping criterion
12  $\mathbf{y}^b := \arg \min_{\{\mathbf{y} \in P \mid \forall \mathbf{z} \in P: f(\mathbf{z}) \not\prec_p f(\mathbf{y})\}} \Psi(f(\mathbf{y}))$ 
13 return  $\mathbf{y}^b$ 

```

---

The performance of the search operators mainly depend on the distribution indices  $\eta$  used with SBX and polynomial mutation. As explained in Section 5.3.1, increasing the distribution index leads to offspring and mutated candidate solutions being located closer to their parents. This in turn dictates whether the search is performed on a local or a more global scale. For this reason, different values for the distribution indices are tested in the study. Note that the same value is chosen for the distribution indices of crossover and mutation, since both operators affect the scope of the search – i.e. distance of parent to offspring individuals – in the same manner.

Hill climbers (HCs) are a family of local search algorithms that iteratively improve a single candidate solution. A random candidate solution is chosen as incumbent. In each iteration, a single new candidate solution – the challenger – is generated by applying a local search operator. If the challenger is better than the incumbent, it supersedes the incumbent [ES<sup>+</sup>03].

---

**Algorithm 16:** hill climber (HC) cf. [ES<sup>+</sup>03]
 

---

**Input** : Cluster  $C = \{\mathbf{x}^i\}_{i=1}^N$ , MOOP  $f$ , scalarization function  $\Psi$ , parameter  $\eta$

**Output:** Local scalarization optimizer approximation  $\mathbf{y}$

```

1  $\mathbf{y} := \arg \min \Psi(\mathbf{f}^i)$ 
2 repeat
3    $\mathbf{z} := \text{polynomialMutation}(\mathbf{y}, \eta)$ 
4   if  $(f(\mathbf{z}) \prec_p f(\mathbf{y})) \vee ((f(\mathbf{y}) \not\prec_p f(\mathbf{z})) \wedge (\Psi(f(\mathbf{z})) < \Psi(f(\mathbf{y}))))$  then
5      $\mathbf{y} := \mathbf{z}$ 
6 until stopping criterion
7 return  $\mathbf{y}$ 

```

---

Algorithm 16 shows an outline of the HC implementation that was used in the study. The cluster member having the smallest scalarization value is chosen as incumbent. Polynomial mutation is applied as local search operator. The challenger replaces the incumbent if it either dominates the incumbent or both objective vectors are non-dominated to each other and the challenger possesses a smaller scalarization value.

The HC may be perceived as a simplified elitist GA that uses a population size of 1 and no crossover operator. If the Pareto front approximation is uniform, the cluster member possessing the smallest scalarization value is expected to be already closely located to the local scalarization optimum of the corresponding basin. A HC might therefore quickly converge to the optimum. The study examines, whether the extra effort of maintaining a population of candidate solutions in the GA is beneficial for the search in the context of local scalarization optima. Different values for the distribution index  $\eta$  of the polynomial mutation are tested with the HC.

The final local search algorithm that is considered in the study is a PSA (Algorithm 17). Single-objective PSAs follow the same algorithmic approach that is utilized within SMPSO (Algorithm 7). Each swarm member  $i$  is associated with a position  $\mathbf{x}^i$  in the decision space and a velocity  $\mathbf{z}_i$ . A particle's position is updated using the same formula as in SMPSO (Equations (5.18) and (5.19) and Lines 7 and 8), however the parameters  $c_1$  and  $c_2$  in Equation (5.19) and the constriction coefficient Equation (5.20) have been dropped in favor of a constriction mechanism that depends solely on the inertia weight  $w$  for simplifying experimentation. This simplification does not restrict the search capabilities of the PSA, since the constriction coefficient and the inertia weight are algebraically equivalent [PKB07]. In dropping  $c_1$ ,  $c_2$  and  $\chi$ , turbulence factors that broaden the search across the search eliminated. This is an intended behavior, to make

## 6. Multimodal Scalarized Preferences

the swarm stay in its assigned basin.

---

**Algorithm 17:** particle swarm algorithm (PSA) cf. [Sia16]

---

**Input :** Cluster  $C = \{\mathbf{x}^i\}_{i=1}^N$ , MOOP  $f$ , scalarization function  $\Psi$ , parameter  $w$

```

1  $S := \{\mathbf{x}^i\}_{i=1}^N$  // swarm
2  $A := S$  with  $A := (\mathbf{y}^1, \dots, \mathbf{y}^N)$  // archive
3  $Z := (\mathbf{z}^1, \dots, \mathbf{z}^N)$  with  $\mathbf{z}^i = \mathbf{0}^n$  // velocities
4  $\mathbf{y}^g := \arg \min \Psi(\mathbf{f}^i)$ 
5 repeat
6   forall  $\mathbf{x}^i \in S$  do
7      $\mathbf{z}^i := w\mathbf{z}^i + r_1(\mathbf{y}^i - \mathbf{s}^i) + r_2(\mathbf{x}^g - \mathbf{s}^i)$ 
8      $\mathbf{x}^i := \mathbf{x}^i + \mathbf{z}^i$ 
9     if  $(\mathbf{f}^i \prec_p f(\mathbf{y}^i)) \vee ((f(\mathbf{y}^i) \prec_p \mathbf{f}^i) \wedge (\Psi(\mathbf{f}^i) < \Psi(f(\mathbf{y}^i))))$  then
10    |  $\mathbf{y}^i := \mathbf{x}^i$ 
11    | if  $(\mathbf{f}^i \prec_p \mathbf{f}^g) \vee ((\mathbf{f}^g \prec_p \mathbf{f}^i) \wedge (\Psi(\mathbf{f}^i) < \Psi(\mathbf{f}^g)))$  then
12    | |  $\mathbf{x}^g := \mathbf{s}^i$ 
13 until stopping criterion
14 return  $\mathbf{x}^g$ 

```

---

The swarm leader is the candidate solution that has the smallest scalarization value among all encountered particles that is not dominated by any swarm member that was generated throughout the search. The personal best position  $\mathbf{y}^i$  of particle  $i$  is the decision vector of the location at which  $i$  has possessed the smallest scalarization value and was not dominated by any previous positions of  $i$ .

PSAs are known to exhibit erratic movements throughout the decision space if the particle velocity is not properly controlled [PKB07]. Such movements also lead to the exploration of vast areas of the search space, effectively leading the PSA to escape its assigned basin. The inertia weight plays an important role in guiding the trajectory of the swarm towards the local optimum. The experimental study therefore assesses how different choices for the value of the inertia weight affects the convergence of the algorithm.

### 6.3.2.2. Local Optimization Performance Indicators

Performance evaluation of local search in single-objective MMO is guided by two aspects. Firstly, any SOO technique aims to minimize the distance between

### 6.3. Computational Analysis of Framework Components

the optimum and the approximation the algorithm computes. Correspondingly, local search in single-objective MMO should approximate local scalarization optima as close as possible. Secondly, each run of the local search should approximate the local optimum of the basin (in the form of the cluster used as initial population) it is assigned to. A local search algorithm that is prone to escaping its assigned basin, in favor of converging to scalarization optima possessing smaller scalarization optima, is expected to be unable to find all local scalarization optima. Both aspects are covered by the performance indicators presented in the following that have been adopted from single-objective MMO by applying a minor modification. In single objective MMO distances are computed in the decision space. The performance indicators in this study all measure distances for performance evaluation in the objective space. All distances are computed in the normalized objective space (Definition 52).

For formally defining the performance indicators, let  $P$  denote the set of local scalarization optima and  $S$  the set of objective vectors obtained through the application of local search. The peak distance (PD) (Definition 66) measures the average distance between each local scalarization optimum  $\mathbf{u}^L$  to its closest approximation in  $S$ . The indicator therefore shows how well each scalarization optimum is approximated on average. A PD of 0 indicates that an exact representation of each optimum was found. PD is algebraically equivalent to the IGD (Definition 57), which measures the quality of Pareto front approximations.

**Definition 66** (peak distance cf. [Pre15]). *Let  $S, P \subset \mathbb{R}^m$  be finite. The peak distance (PD)  $PD(S, P)$  is defined as*

$$PD(S, P) = \frac{1}{|P|} \sum_{\mathbf{u}^L \in P} \min_{\mathbf{v} \in S} \|\mathbf{u}^L - \mathbf{v}\|_2. \quad (6.5)$$

Performance measured by PD is prone to the influence of outliers. If a single scalarization optimum is missed during local search – for example because the local search escaped its assigned basin – the PD may greatly deteriorate even if an exact representation of all other optima is found.

Often, it is sufficient to approximate an optimum within a certain accuracy, since in practical applications, decision variable values may only be set within a limited precision [Pre15]. In such cases, it suffices that the approximation lies within a given distance threshold of the local optimum. The preak ratio (PR) (Definition 67) follows this notion by counting the number of optima for which there exists an approximation within a distance of  $\varepsilon \leq 0$  divided by the total number of optima. An indicator value of 1 states that each optimum has been approximated by a point that lies within a range of  $\varepsilon$ .

## 6. Multimodal Scalarized Preferences

**Definition 67** (peak ratio cf. [Pre15]). *Let  $S, P \subset \mathbb{R}^m$  be finite and  $\varepsilon \in \mathbb{R}_+$ . The peak ratio (PR)  $PR(S, P)$  is defined as*

$$PR(S, P) = \frac{1}{|P|} \sum_{\mathbf{u}^L \in P} 1_{\exists \mathbf{v} \in S: \|\mathbf{u}^L - \mathbf{v}\|_2 \leq \varepsilon}. \quad (6.6)$$

The PR can be interpreted as indicator for how many optima are found during local search. The indicator, however, makes no statement about the quality of their approximation. The expressiveness of the PR depends on the chosen threshold  $\varepsilon$ . If  $\varepsilon$  is chosen too large, the optima are too coarsely approximated. If  $\varepsilon$  is chosen too small, there may be only few optima for which there exists a close enough approximation. A value of  $\varepsilon = 0.01$  in the normalized objective space was chosen for the study.

The final performance indicator is a measure that originates in statistics. A false positive (FP) is an objective vector that has been identified as scalarization optimum by local search although it is located too far away from a true optimum. The FP indicator (Definition 68) counts the number local optima approximations that do not lie within a threshold  $\delta$  of any scalarization optimum. Similar to PR, the expressiveness of FP depends on the chosen value for  $\delta$  and is subject to the opinion of the DM. A value of  $\delta = 0.2$  was chosen for the study.

**Definition 68** (false positives cf. [Pre15]). *Let  $S, P \subset \mathbb{R}^m$  be finite and  $\delta \in \mathbb{R}_+$ . The false positive (FP)  $FP(S, P)$  performance indicator is defined as*

$$FP(S, P) = \sum_{\mathbf{v} \in S} 1_{\forall \mathbf{u}^L \in P: \|\mathbf{u}^L - \mathbf{v}\|_2 > \delta}. \quad (6.7)$$

FPs are particularly undesirable from a decision-making perspective, since there exist other candidate solutions in the proximity of an FP that possess a smaller scalarization value. If the search is too narrow, FPs may occur. Another reason for the occurrence of FP may be insufficient information about reference points if the scalarization function violates binary independence (Definition 45). In this case, scalarization values are computed incorrectly, which leads local search to assume local optima at other positions than where they are truly located. Even if the local optimization succeeds in this case, the approximated optima are not of any use.

### 6.3.2.3. Discussion of the Local Optimization Study Results

A summary of the experimental settings is found in the appendix (Table E.5). Preliminary testing was conducted and recommended values from the literature were considered to identify meaningful ranges for the parameters of the

### 6.3. Computational Analysis of Framework Components

local optimization algorithms. Hansen recommends choosing a step size in the interval  $[1e-3, 1e-1]$  if the feasible set is contained in a bounding box with side length 10 to commence the optimization by a narrow local search.<sup>4</sup> The feasible set of all problems considered in the study is given by  $X = [0, 1]^n$ . Therefore, the step sizes  $\sigma \in \{1e-5, 1e-4, 1e-3, 1e-2, 1e-1\}$  were chosen for the study.

A distribution index of  $\eta = 20$  is commonly used in the literature for both SBX and polynomial mutation to maintain a balance between exploration and exploitation during the search [DPAM02, ZLT01, ZK04]. Consequently, the distribution index should be increased to focus the search narrowly inside the basin. The values  $\eta \in \{20, 30, 40, 80, 160\}$  were chosen to take into account that local optimization needs to put a stronger focus on exploitation than exploration.

Since the constriction factor and the turbulence mechanism are removed from Algorithm 17, the inertia weight is the driving force of the search that determines the magnitude of the particles' movement in the search space. Initial tests have revealed that inertia weights smaller than 0.1 result in movements that are too small to provide substantial improvements during the local search. Inertia weights greater than 1 are ill-suited from an algebraic perspective. Line 7 in Algorithm 17 pulls the particle towards its personal and the global best position. If  $w > 1$  then Line 7 may result in pulling the particle beyond the position of the local and global best. Therefore, values between 0.1 and 1 were chosen for the study:  $w \in \{0.1, 0.25, 0.5, 0.75, 1\}$ .

Tables G.1 to G.4 show median PDs across 100 runs. The step size  $\sigma = 1e-3$  consistently delivers the best results across all four scalarization functions. The parametrization  $\sigma = 1e-2$  achieves the second best performance for the sum of objectives, the Nash bargaining solution and angle utility. For tradeoff utility,  $\sigma = 1e-4$  results in smaller PDs than  $\sigma = 1e-2$ . Since tradeoff utility landscapes of the chosen test problems are highly multimodal and possess several local scalarization optima, the narrower search focus induced by the smaller step size appears to benefit the local search.

There exist notable differences in the reported PDs between the five parametrizations up to a magnitude of  $10e3$ . On average, however, all chosen step sizes appear to result in successful approximations of the local scalarization optima. There exist a few notable exceptions, though, for the step sizes  $1e-5$  and  $1e-4$ : On a number of DO2DK, DTLZ7 and ZDT3 test instances all approximated minima obtained by CMA-ES using the step sizes  $1e-5$  and  $1e-4$  are far away from the the true minima. In these cases, the search may be too narrow to converge towards the scalarization optima.

<sup>4</sup>[https://www.lri.fr/~hansen/cmaes\\_inmatlab.html#practical](https://www.lri.fr/~hansen/cmaes_inmatlab.html#practical).  
23.10.2017.

Accessed

## 6. Multimodal Scalarized Preferences

The PDs reported for the GA and the HC exhibit a similar performance pattern with respect to the distribution index. Increasing the distribution index steadily improves the results suggesting that local optimization benefits from a narrow search focus. Preliminary tests have revealed that higher distribution indices are able to achieve even smaller PDs. Further increasing the distribution index, however, is questionable from an algebraic perspective, since offspring and mutated individuals would be too close to their parents to be able to effectively explore the basin. The observed results rather suggest that the clusters possess elements that are already close to the scalarization minima. A search that is very narrow may achieve good results in this situation, since small, but steady improvements can lead to a sufficient approximation of the local optimum. If the initial approximation of the optimum is rather coarse, however, large distribution indices are expected to fail in finding a close approximation of the optimum.

A closer look at the figures in Tables G.2 and G.3 also reveal that the performance increase between the distribution index 80 and 120 becomes small to negligible on most problem instances. This observation further suggests that increasing the distribution index beyond 120 will not provide a significant improvement of the results. The tendency of attaining smaller PDs by increasing the distribution index appears to be stronger for the GA compared to the HC. Choosing a distribution index of 120 for the GA results in the smallest PD on nearly all test problems. The HC shows more heterogeneous results, especially for tradeoff utility. Although the HC with a distribution index 120 still achieves the best performance on average, other parametrization outperform  $\eta = 120$  on multiple problems. The performance difference on these problem instances on which the parametrization  $\eta = 120$  is outperformed, however, is negligible, further evidencing that increasing  $\eta$  beyond 120 is not entirely beneficial.

The results for the PSO identify the inertia weight  $w = 0.75$  as the best parametrization. The choice  $w = 0.75$  outperforms all other parametrizations on the majority of the considered problems. The parametrization  $w = 0.5$  attains the second best performance. The results suggest that personal and global best should exhibit a strong influence in computing particle velocity. At the same time, increasing the inertia weight beyond 0.75 leads to a notable performance drop. The performance decrease from  $w = 0.75$  to  $w = 0.1$  is far less pronounced.

Tables G.5 to G.8 show the algorithms' performances with respect to the PR. In setting a threshold for the detection of local scalarization optima, the PR defines a minimum aspiration level for each optimization algorithm. Every parametrization that achieves a PR of 1 is deemed successful in approximating the local scalarization optima of the given problem.

### 6.3. Computational Analysis of Framework Components

The results confirm that a step size of  $\sigma = 1e-2$  allows CMA-ES to find good approximations of all scalarization optima on the majority of the considered test problems. The step size  $\sigma = 1e-2$  leads CMA-ES to achieve the highest PRs among all parametrizations. The parametrizations  $\sigma \in \{1e-5, 1e-4\}$ , on the other hand, perform rather poorly, not finding sufficient approximations of multiple local optima on many of the test problems. The PR for the step size  $1e-3$  is only smaller on a few test problems compared to  $\sigma = 1e-2$ . The step size  $1e-1$  also generates good results with the exception of the sum of objectives scalarization function. In general, a recommendation for both  $\sigma \in \{1e-3, 1e-2\}$  can be given.

The distribution indices  $\eta \in \{80, 120\}$  result in a similar performance with respect to the PR for both the GA and the HC. Smaller distribution indices are discouraged by the results, since performance drops for  $\eta < 80$  are observed for the sum of objectives and tradeoff utility. The figures suggest that small distribution indices broaden the search too far such that the local optimization algorithm escapes its assigned basin.

The PSA with inertia weights in the range  $w \in \{0.1, 0.25, 0.5, 0.75\}$  yields good results with respect to the PR across all scalarization functions. The parametrization  $w = 0.75$  is best among all tested values. A sharp performance drop can be observed for  $w = 1$ . On the majority of the considered test problems, only a fraction of all scalarization optima is sufficiently approximated if  $w = 1$ . This observation suggests that there exists a threshold value for the inertia weight beyond which the approximation quality quickly declines.

Tables G.9 to G.12 show the FPs obtained by the different parametrizations of the algorithms. Most false positives occur on the DEB3DK  $k = 2$  (tradeoff utility) and DTLZ7 (angle and tradeoff utility) problems. A closer analysis has revealed, however, that the overall majority of these FPs is located in the vicinity of true local scalarization optima. The FPs are still too far away from the actual minima, though, which is why they are identified as FPs. In this sense, the reported FPs should be rather interpreted as coarse approximations of local scalarization optima.

The best parametrizations that have been identified during the analysis of PD and PR ( $\sigma = 1e-3$ ,  $\eta = 120$ ,  $w = 0.75$ ) obtain little to no FPs. These are, however, not the best parametrizations if only FPs are considered. The step size  $\sigma = 1e-2$  results in fewer FPs than  $\sigma = 1e-3$ . The PSA also obtains the least FPs with inertia weights  $w \in \{0.1, 0.25\}$ .

All in all, the meaning of FPs should not be overstated. While coarse approximations are a nuisance, PR and PD carry more weight in identifying good parametrizations. High PRs and small PDs indicate that a local optimization

## 6. Multimodal Scalarized Preferences

algorithm is capable of finding good approximations of local scalarization optima. The occurrence of FPs rather implies that the approximations are sometimes too coarse. The results also point to an inherent tradeoff between PD or PR and FPs. Fine-tuning the parametrization of a local search algorithm can lead to the identification of more local optima. It may happen though that these additional optima are only coarsely approximated, since they are in general hard to find.

The FPs reported for the GA and the HC, however confirm the tendency of previous findings with respect to the PD and PR. Increasing the distribution index, unanimously leads to the identification of fewer FPs. Small distribution indices appear to broaden the search too far, such that both algorithms do not converge properly.

The final part of the analysis consists of a comparison of the best parametrization of each algorithm. Table 6.7 shows the PDs of CMA-ES ( $\sigma = 1e-3$ ), GA, HC (each  $\eta = 120$ ) and the PSA ( $w = 0.75$ ). The comparison also includes a baseline to assess the effectiveness of the local search. The baseline identifies the objective vector of each cluster that possesses the smallest scalarization value as estimate of a local scalarization optimum. Thereby, the baseline shows the approximation quality of the local optima obtained by the Pareto front approximation without any local optimization effort.

A convergence analysis as performed in Section 5.3 is omitted, since the performance of a single local optimization run is computed from the execution of multiple local searches. Each local search is aimed at approximating a different local optimum making the analysis of the presented performance indicators across multiple iterations ambiguous. For example, a local search that converges towards a specific local optimum is likely to move further away from other local optima at the same time. If none of the basins of these optima have been found by the clustering, for example, the PD might decrease although the local search is successful in approximating the optimum of the assigned basin. The subsequent analysis also focuses solely on the PD, since the best parametrizations of each algorithm exhibited similar results for both the PR and FP.

The results in Table 6.7 show that the HC and the PSA are the top performing local optimization algorithms. There exist notable performance differences between the four scalarization functions. The PSA is superior to the HC on the sum of objectives and the Nash bargaining solution problems. The HC outperforms the PSA on the angle utility instances. On the tradeoff utility problems, both algorithms exhibit a similar performance. CMA-ES and the GA should be considered outperformed by the HC and the PSA. The baseline also achieves the smallest PD on three and the second smallest on one problem instance.

### 6.3. Computational Analysis of Framework Components

This observation suggests that the local optimization algorithms escape their assigned basin during the search.

A statistical analysis is carried out to confirm, whether the observed performance differences are significant. The same procedure that is applied in Chapter 5 is used to evaluate the simulation results. The distribution of the PDs of each algorithm on every problem is tested for normality. Subsequent ANOVA or Kruskal-Wallis tests are applied on each test problem to confirm, whether there exists at least one algorithm, whose performance significantly differs from those of the other algorithms. The test results confirm that such significant performance differences exist on every test problem (Table G.18)

Table 6.8 shows a post-hoc comparison of the baseline to the local optimization algorithms. The figures indicate that the local optimization leads to a significant improvement of the approximations of scalarization optima on the majority of the test problems. The results, however, also confirm that performance deteriorates in comparison to the baseline on a few benchmark problems. The baseline outperforms all four algorithms with confidence on the ZDT3 Nash bargaining solution and on the DTLZ7 angle utility instance.

It must be noted, however, that this does not imply that the optimization itself fails. Instead, the performance deterioration is either attributed to the local optimization algorithms escaping their dedicated basins or the clustering not detecting all basins. The former case corresponds to an undesirable behavior that should be avoided by the algorithm. Since the algorithm neither knows nor explicitly uses the basin structure of the problem, it converges to an optimum that possesses a smaller scalarization value than the one of its assigned basin. Although this behavior is not intended, it is not a failure of the algorithm's working principles.

The latter case – the clustering not detecting all basins – should be interpreted as a conceptual shortcoming of the PD. As explained before, a point that moves towards a specific local optimum might move away from other local optima at the same time, which may increase the PD. If a cluster covers multiple basins, the member of the cluster that possesses the smallest scalarization value might possess a smaller average distance to the local optima of the covered basins than a point obtained through local search.

The four algorithms and the baseline are ranked to deliver a final verdict on their performance. The same methodology that is applied to evaluate ESPEA against other finite set of points algorithms is used. For each problem instance and algorithm a rank is computed. The rank consists of the number of algorithms that perform significantly worse with respect to the PD than the considered algorithm (Table G.22). The average ranks across all problems and scalarization functions are displayed in Table 6.9. The figures show that using either

## 6. Multimodal Scalarized Preferences

the PSA, HC or CMA-ES results in a significant improvement of the PD. In conjunction with the median PDs displayed in Table 6.8, it is possible to conclude that local optimization results in a large improvement of the approximations of the local scalarization optima on the vast majority of the test problems. The GA is the only algorithm that does not achieve a significant improvement at a 95 % confidence level. Since the observed  $p$ -value is close to 0.05, however, a performance improvement is still likely to occur.

Table 6.10 shows the complete results of the post-hoc analysis carried out to compare the ranks of the algorithms. The figures show that the PSA significantly outperforms the GA and CMA-ES. The performance difference between the PSA and the HC is negligible. The HC outperforms the GA with confidence, however not CMA-ES. The performance difference between the GA and CMA-ES is not significant.

Table 6.7.: Local search study – PD. Median and IQR (as subscript) results of best local search parametrizations:  $\sigma = 1e-3$  (CMA-ES),  $\eta = 120$  (GA and HC) and  $w = 0.75$  (PSA). The baseline shows performance before any local optimization. Each objective vector possessing the smallest scalarization value in every cluster was chosen as estimate for a local scalarization minimum. Best performances are colored in green, second-best performances in blue. The HC and PSA are the top performing algorithms.

	Sum of objectives					Nash				
	Baseline	CMA-ES	GA	HC	PSA	Baseline	CMA-ES	GA	HC	PSA
	DEB2DK $k = 1$	3.09e-3 <sub>1.42e-3</sub>	3.46e-4 <sub>4.06e-4</sub>	2.08e-3 <sub>7.07e-4</sub>	1.26e-5 <sub>2.20e-5</sub>	1.80e-9 <sub>1.09e-9</sub>	3.48e-3 <sub>3.74e-3</sub>	1.18e-3 <sub>1.36e-3</sub>	1.78e-3 <sub>1.89e-3</sub>	4.62e-5 <sub>7.98e-5</sub>
DEB2DK $k = 2$	3.61e-3 <sub>3.35e-3</sub>	7.97e-4 <sub>5.65e-4</sub>	2.74e-3 <sub>1.90e-3</sub>	7.04e-5 <sub>1.25e-1</sub>	3.67e-9 <sub>5.58e-10</sub>	4.42e-3 <sub>2.77e-3</sub>	1.31e-3 <sub>1.04e-3</sub>	2.91e-3 <sub>1.07e-3</sub>	1.20e-4 <sub>8.06e-5</sub>	9.88e-5 <sub>1.50e-5</sub>
DEB2DK $k = 3$	3.80e-3 <sub>1.28e-3</sub>	9.18e-4 <sub>5.26e-4</sub>	7.37e-2 <sub>5.73e-2</sub>	1.24e-1 <sub>7.24e-2</sub>	4.04e-9 <sub>6.11e-10</sub>	4.30e-3 <sub>1.95e-3</sub>	1.37e-3 <sub>9.32e-4</sub>	3.14e-3 <sub>1.46e-3</sub>	1.67e-4 <sub>9.16e-5</sub>	1.30e-4 <sub>2.38e-5</sub>
DEB2DK $k = 4$	3.80e-3 <sub>1.44e-3</sub>	8.73e-4 <sub>4.92e-4</sub>	9.32e-2 <sub>2.25e-2</sub>	9.35e-2 <sub>8.72e-5</sub>	5.20e-9 <sub>3.63e-10</sub>	5.23e-3 <sub>1.89e-3</sub>	2.03e-3 <sub>1.63e-3</sub>	5.53e-3 <sub>6.48e-2</sub>	3.73e-2 <sub>6.70e-2</sub>	1.34e-1 <sub>1.79e-7</sub>
DEB3DK $k = 1$	4.92e-2 <sub>9.72e-2</sub>	2.70e-1 <sub>1.73e-1</sub>	8.09e-3 <sub>1.11e-1</sub>	1.14e-4 <sub>1.13e-1</sub>	1.73e-4 <sub>1.13e-1</sub>	2.92e-2 <sub>1.26e-2</sub>	1.98e-2 <sub>5.76</sub>	3.75e-3 <sub>2.79e-3</sub>	3.15e-3 <sub>6.45e-4</sub>	3.27e-3 <sub>7.02e-4</sub>
DEB3DK $k = 2$	6.03e-2 <sub>2.85e-2</sub>	9.76e-2 <sub>6.39e-2</sub>	4.13e-2 <sub>3.80e-2</sub>	1.57e-2 <sub>2.41e-2</sub>	1.60e-2 <sub>3.75e-2</sub>	6.71e-2 <sub>2.73e-2</sub>	3.30e-1 <sub>5.22</sub>	4.25e-2 <sub>3.87e-2</sub>	4.26e-2 <sub>3.83e-2</sub>	4.26e-2 <sub>3.78e-2</sub>
DO2DK $k = 1 s = 1$	2.58e-3 <sub>3.04e-3</sub>	7.05e-4 <sub>8.89e-4</sub>	1.19e-3 <sub>1.49e-3</sub>	2.74e-5 <sub>6.04e-5</sub>	5.31e-9 <sub>1.17e-9</sub>	2.44e-3 <sub>2.59e-3</sub>	5.31e-4 <sub>6.49e-4</sub>	1.31e-3 <sub>1.25e-3</sub>	2.70e-5 <sub>8.82e-5</sub>	1.66e-5 <sub>2.19e-5</sub>
DO2DK $k = 2 s = 1$	5.09e-3 <sub>4.58e-3</sub>	1.87e-3 <sub>2.07e-3</sub>	3.16e-3 <sub>2.41e-3</sub>	6.78e-5 <sub>7.80e-5</sub>	1.50e-9 <sub>1.60e-10</sub>	3.21e-3 <sub>2.16e-3</sub>	7.90e-4 <sub>8.52e-4</sub>	2.26e-3 <sub>1.50e-3</sub>	6.80e-5 <sub>8.85e-5</sub>	4.43e-5 <sub>1.02e-5</sub>
DO2DK $k = 3 s = 1$	3.38e-3 <sub>1.87e-3</sub>	8.88e-4 <sub>6.69e-4</sub>	2.19e-3 <sub>1.50e-3</sub>	6.88e-5 <sub>7.04e-5</sub>	5.67e-9 <sub>1.09e-9</sub>	2.49e-3 <sub>2.31e-3</sub>	6.08e-4 <sub>6.96e-4</sub>	1.56e-3 <sub>1.48e-3</sub>	4.99e-5 <sub>8.85e-5</sub>	2.98e-5 <sub>1.96e-5</sub>
DO2DK $k = 4 s = 1$	5.38e-3 <sub>9.18e-2</sub>	1.69e-3 <sub>9.35e-2</sub>	9.36e-2 <sub>9.22e-2</sub>	9.39e-2 <sub>9.39e-2</sub>	9.39e-2 <sub>7.78e-10</sub>	3.08e-3 <sub>1.48e-3</sub>	6.03e-4 <sub>4.93e-4</sub>	2.08e-3 <sub>9.83e-4</sub>	6.46e-5 <sub>5.53e-5</sub>	2.05e-5 <sub>1.55e-5</sub>
DTLZ7	3.92e-2 <sub>1.09e-2</sub>	2.10e-1 <sub>4.03e-1</sub>	8.78e-3 <sub>2.65e-3</sub>	7.12e-3 <sub>7.31e-5</sub>	7.12e-3 <sub>1.17e-9</sub>	5.13e-2 <sub>1.58e-2</sub>	1.72e-1 <sub>1.53e-1</sub>	1.98e-2 <sub>8.93e-3</sub>	2.36e-2 <sub>8.57e-3</sub>	2.22e-2 <sub>9.28e-3</sub>
ZDT3	4.74e-3 <sub>1.17e-3</sub>	2.42e-4 <sub>1.99e-4</sub>	1.61e-3 <sub>4.85e-4</sub>	9.69e-5 <sub>5.79e-5</sub>	6.27e-9 <sub>1.89e-10</sub>	4.41e-3 <sub>1.59e-3</sub>	4.10e-2 <sub>1.58e-3</sub>	4.21e-2 <sub>2.21e-3</sub>	4.08e-2 <sub>1.47e-3</sub>	4.08e-2 <sub>1.32e-3</sub>
	Angle utility					Tradeoff utility				
	Baseline	CMA-ES	GA	HC	PSA	Baseline	CMA-ES	GA	HC	PSA
DEB2DK $k = 1$	2.40e-3 <sub>1.20e-3</sub>	1.34e-4 <sub>1.73e-4</sub>	1.24e-3 <sub>5.30e-4</sub>	1.36e-4 <sub>1.70e-4</sub>	2.78e-4 <sub>6.63e-4</sub>	3.09e-3 <sub>1.42e-3</sub>	2.11e-3 <sub>4.90e-4</sub>	2.24e-3 <sub>5.21e-4</sub>	1.24e-4 <sub>2.27e-4</sub>	1.75e-4 <sub>2.00e-4</sub>
DEB2DK $k = 2$	4.53e-3 <sub>2.23e-3</sub>	1.53e-3 <sub>1.03e-3</sub>	3.15e-3 <sub>1.89e-3</sub>	1.81e-4 <sub>5.75e-5</sub>	1.88e-4 <sub>2.89e-5</sub>	3.72e-3 <sub>1.21e-3</sub>	1.62e-3 <sub>3.62e-4</sub>	2.34e-3 <sub>6.76e-4</sub>	2.40e-4 <sub>9.92e-5</sub>	3.01e-4 <sub>1.39e-4</sub>
DEB2DK $k = 3$	3.81e-3 <sub>1.71e-3</sub>	1.21e-3 <sub>8.40e-4</sub>	2.49e-3 <sub>1.17e-3</sub>	2.82e-4 <sub>1.98e-4</sub>	2.94e-4 <sub>2.16e-4</sub>	3.67e-3 <sub>3.60e-4</sub>	2.30e-3 <sub>8.60e-4</sub>	2.40e-3 <sub>6.40e-4</sub>	1.19e-3 <sub>5.53e-4</sub>	1.04e-3 <sub>3.24e-4</sub>
DEB2DK $k = 4$	4.10e-3 <sub>1.69e-3</sub>	1.37e-3 <sub>8.85e-4</sub>	2.97e-3 <sub>9.76e-4</sub>	1.59e-4 <sub>2.9e-5</sub>	1.42e-4 <sub>2.54e-5</sub>	3.80e-3 <sub>9.92e-4</sub>	2.21e-3 <sub>3.74e-4</sub>	2.57e-3 <sub>8.05e-4</sub>	1.64e-2 <sub>2.45e-2</sub>	1.11e-3 <sub>5.52e-4</sub>
DEB3DK $k = 1$	3.53e-2 <sub>1.14e-2</sub>	3.43e-2 <sub>2.84e-2</sub>	3.73e-3 <sub>3.27e-3</sub>	1.46e-3 <sub>3.19e-4</sub>	1.50e-3 <sub>3.41e-4</sub>	6.49e-2 <sub>1.57e-2</sub>	5.88e-2 <sub>2.12e-2</sub>	4.53e-2 <sub>2.33e-2</sub>	4.56e-2 <sub>1.31e-2</sub>	4.40e-2 <sub>1.57e-2</sub>
DEB3DK $k = 2$	6.54e-2 <sub>1.03e-2</sub>	3.13e-1 <sub>1.86e-1</sub>	3.96e-2 <sub>2.65e-2</sub>	4.10e-2 <sub>2.72e-2</sub>	4.12e-2 <sub>3.30e-3</sub>	1.19e-1 <sub>1.64e-2</sub>	1.20e-1 <sub>1.76e-2</sub>	1.16e-1 <sub>1.69e-2</sub>	1.16e-1 <sub>8.8e-2</sub>	1.21e-1 <sub>1.98e-2</sub>
DO2DK $k = 1 s = 1$	1.97e-3 <sub>2.12e-3</sub>	1.55e-4 <sub>1.90e-4</sub>	9.34e-4 <sub>4.65e-4</sub>	1.12e-4 <sub>2.17e-4</sub>	2.21e-4 <sub>3.30e-3</sub>	3.11e-3 <sub>1.64e-3</sub>	3.97e-4 <sub>1.76e-2</sub>	2.13e-3 <sub>1.35e-3</sub>	1.25e-4 <sub>3.02e-4</sub>	2.00e-4 <sub>43.00e-4</sub>
DO2DK $k = 2 s = 1$	4.24e-3 <sub>3.63e-3</sub>	9.58e-5 <sub>5.59e-5</sub>	1.29e-3 <sub>3.45e-4</sub>	7.18e-5 <sub>2.35e-5</sub>	1.80e-4 <sub>8.11e-4</sub>	3.59e-3 <sub>3.60e-3</sub>	1.79e-3 <sub>2.7e-3</sub>	1.91e-3 <sub>7.49e-4</sub>	5.20e-4 <sub>3.91e-4</sub>	5.84e-4 <sub>3.17e-4</sub>
DO2DK $k = 3 s = 1$	6.17e-3 <sub>3.40e-3</sub>	3.27e-4 <sub>1.46e-4</sub>	1.66e-3 <sub>9.69e-4</sub>	2.15e-4 <sub>2.25e-4</sub>	3.74e-4 <sub>4.17e-4</sub>	3.86e-3 <sub>3.28e-3</sub>	3.58e-3 <sub>2.7e-3</sub>	2.30e-3 <sub>1.06e-3</sub>	1.52e-2 <sub>3.45e-4</sub>	1.45e-2 <sub>1.29e-2</sub>
DO2DK $k = 4 s = 1$	2.50e-3 <sub>2.31e-3</sub>	7.12e-4 <sub>4.36e-4</sub>	1.29e-3 <sub>1.02e-3</sub>	8.71e-4 <sub>5.85e-4</sub>	6.62e-4 <sub>5.44e-4</sub>	4.04e-3 <sub>1.72e-3</sub>	1.88e-3 <sub>1.15e-3</sub>	2.26e-3 <sub>8.39e-4</sub>	5.72e-4 <sub>2.20e-4</sub>	6.41e-4 <sub>2.15e-4</sub>
DTLZ7	4.68e-2 <sub>2.10e-2</sub>	2.26e-1 <sub>6.38e-2</sub>	1.79e-1 <sub>1.35e-2</sub>	1.95e-1 <sub>1.22e-2</sub>	1.85e-1 <sub>6.23e-3</sub>	7.23e-2 <sub>2.19e-2</sub>	8.07e-2 <sub>2.90e-2</sub>	1.64e-1 <sub>2.54e-2</sub>	1.63e-1 <sub>3.34e-2</sub>	1.15e-1 <sub>3.31e-2</sub>
ZDT3	5.68e-3 <sub>1.57e-3</sub>	1.91e-3 <sub>1.90e-4</sub>	2.72e-3 <sub>6.90e-4</sub>	1.84e-3 <sub>1.04e-4</sub>	1.85e-3 <sub>9.00e-6</sub>	4.66e-3 <sub>3.37e-3</sub>	1.27e-3 <sub>3.41e-4</sub>	2.49e-2 <sub>2.38e-2</sub>	2.48e-2 <sub>6.98e-5</sub>	1.30e-3 <sub>3.41e-4</sub>

Table 6.8.: Local search study – PD. The table shows  $p$ -values of a post-hoc analysis of the best local search parametrizations. Green cell color indicates that the baseline is outperformed by the corresponding algorithm with confidence at a 95 % level, the blue color without confidence. Red cell color indicates that the baseline is outperformed by the corresponding algorithm with confidence at a 95 % level, the orange color without confidence.

	Sum of objectives				Nash			
	CMA-ES	GA	HC	PSA	CMA-ES	GA	HC	PSA
DEB2DK $k = 1$	0.0000	0.0029	0.0000	0.0000	0.0000	0.0071	0.0000	0.0000
DEB2DK $k = 2$	0.0000	0.7775	0.0000	0.0000	0.0000	0.0584	0.0000	0.0000
DEB2DK $k = 3$	0.0000	0.0000	0.0000	0.0000	0.0000	0.0214	0.0000	0.0000
DEB2DK $k = 4$	0.0000	0.0000	0.0000	0.0000	0.0000	0.9980	0.2669	0.0000
DEB3DK $k = 1$	0.0000	0.0883	0.0000	0.0000	0.0070	0.0000	0.0000	0.0000
DEB3DK $k = 2$	0.0003	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
DO2DK $k = 1 s = 1$	0.0000	0.0112	0.0000	0.0000	0.0000	0.0061	0.0000	0.0000
DO2DK $k = 2 s = 1$	0.0000	0.0224	0.0000	0.0000	0.0000	0.0776	0.0000	0.0000
DO2DK $k = 3 s = 1$	0.0000	0.1100	0.0000	0.0000	0.0000	0.0503	0.0000	0.0000
DO2DK $k = 4 s = 1$	0.0858	0.9965	0.3262	0.9999	0.0000	0.0204	0.0000	0.0000
DTLZ7	0.9956	0.0000	0.0000	0.0000	0.8326	0.0000	0.0000	0.0000
ZDT3	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	Angle utility				Tradeoff utility			
DEB2DK $k = 1$	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
DEB2DK $k = 2$	0.0000	0.0282	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
DEB2DK $k = 3$	0.0000	0.0026	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
DEB2DK $k = 4$	0.0000	0.0021	0.0000	0.0000	0.0000	0.0000	0.8581	0.0000
DEB3DK $k = 1$	0.9991	0.0000	0.0000	0.0000	0.0046	0.0000	0.0000	0.0000
DEB3DK $k = 2$	0.0000	0.0000	0.0000	0.0000	0.9432	0.9648	0.8943	0.3945
DO2DK $k = 1 s = 1$	0.0000	0.0015	0.0000	0.0000	0.0000	0.0012	0.0000	0.0000
DO2DK $k = 2 s = 1$	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
DO2DK $k = 3 s = 1$	0.0000	0.0000	0.0000	0.0000	0.9956	0.0000	0.0000	0.5876
DO2DK $k = 4 s = 1$	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
DTLZ7	0.0000	0.0000	0.0000	0.0000	0.5756	0.0000	0.0000	0.0000
ZDT3	0.0000	0.0000	0.0000	0.0000	0.0000	0.6130	0.0000	0.0000

### 6.3. Computational Analysis of Framework Components

Table 6.9.: Local search study – PD. Friedman tests results with respect to the baseline. The rank states the number of other algorithms an algorithm significantly outperforms on average. The column  $p$ -value is related to the performance difference with respect to the baseline. Green cell color indicates that the baseline outperforms the corresponding algorithm with confidence at a 95 % level, the blue color without confidence. The results allow the conclusion that, in general, the local search further improves the approximation of the local optima. All algorithms besides the GA outperform the baseline with confidence at a 95 % level. The  $p$ -value of the GA is close to 0.05 suggesting that applying the GA as local search still yields a noticeable improvement.

Rank	Algorithm	$p$ -value
3.0417	PSA	1.6098e-13
2.8125	HC	4.0844e-11
1.9688	CMA-ES	2.7361e-4
1.5000	GA	0.0728
0.6771	Baseline	n/a

Table 6.10.: Local search study – PD. Friedman test results in detail. Colors indicate the following: red implies that the row algorithm is outperformed by column algorithm at a 95 % confidence level; orange implies that the row algorithm is outperformed without confidence and the blue color that the row algorithm outperforms the respective column algorithm.

	CMA-ES	GA	HC	PSA
Baseline	2.7361e-4	0.0728	4.0844e-11	1.6098e-13
CMA-ES		0.7457	0.0598	0.0049
GA			2.0257e-4	5.5522e-6
HC				0.9978



## 7. Real-world Applications

Optimization theory is a discipline that is founded in the analysis of real-world applications. Methodologies that are born from entirely theoretical assumptions about reality should therefore be tested on real-world applications to assess their performance and applicability in practice. The scope of this chapter therefore lies in illustrating the usefulness of the algorithms presented in Chapters 5 and 6 with two practical examples.

The first example is concerned with the optimization of the operation efficiency of a CHP plant. ESPEA is applied to generate operating schedules for the CHP plant that maximize fuel efficiency. At the same time, ESPEA finds alternative schedules that are close to but deviate from the efficiency optimum. Such minor deviations from the efficiency optimum are occasionally beneficial to improve the short-term profitability of the plant.

The second example explores optimal scheduling in building energy management systems (BEMSs). Building energy management pursues different, usually conflicting goals in operating buildings such as minimizing costs and carbon dioxide emissions while maximizing user comfort. Building operations consists of scheduling the execution times and modes of household appliances and the control of heating, ventilation and air-conditioning (HVAC) devices. Classic MOO methodology generates a multitude of operating schedules that are often difficult to compare and choose from. Restricting the set of candidate solutions to local scalarization optima can greatly decrease the number of potentially interesting schedules thereby facilitating a simplification of the decision-making process.

### 7.1. Combined Heat and Power Plant

*Declaration: Parts of this section have been published in [BSE<sup>+</sup>16]. Some passages are quoted verbatim.*

Cogeneration is the simultaneous generation of electricity and useful heat with the aim of exploiting more efficiently the energy stored in the fuel. Cogeneration is, however, a complex process that encompasses a great amount of sub-

## 7. Real-world Applications

systems and variables. This fact makes it very difficult to obtain an analytical model of an entire plant, and therefore providing a mechanism or a methodology able to optimize its global behavior.

This section proposes a neuro-evolutionary strategy for modeling and optimizing a cogeneration process of a real industrial plant. Firstly, the modeling of the process is carried out by means of several interconnected artificial neural networks (ANNs), where each ANN deals with a particular sub-system of the plant. Next, the obtained models are used as black box functions by an evolutionary algorithm, which solves a multi-objective optimization problem of the plant. The objectives of the optimization problem consist of minimizing the fuel consumption and maximizing both the generated electricity and the use of the heat. The proposed approach is evaluated with data of a real cogeneration plant collected over a one-year period. Obtained results verify that the modeling of the plant is correct but also that the optimization increases the efficiency of the cogeneration plant.

### 7.1.1. Modeling of a Complex Cogeneration Process

The process of generating electricity and useful heat at the same time is called cogeneration and is also known as combined heat and power. The ultimate goal of cogeneration is to exploit the maximum possible energy contained in a fuel. In the industry, the high temperature flue gases generated by engines, gas turbines, or other machines can be used to produce more electricity or to perform another process demanding heat. This implies cost savings because the amount of fuel required is reduced. This fuel saving also results in a reduction of pollution. These economic and environmental factors are the reasons why nowadays the number of cogeneration plants is increasing steadily.

As many other industrial processes, cogeneration is a rather complex process due to a high number of variables involved, nonlinear dynamics, limited analytical models and also incomplete knowledge. This fact implies that it is highly arduous to obtain a model that reproduces with fidelity the behavior of the real system. Moreover, without such a model, it becomes very difficult to carry out any formal strategy to try to optimize the efficiency of the process. Especially, the application of mathematical programming techniques is impossible if a formal mathematical description cannot be obtained.

Soft computing methods provide a non-conventional way to deal with those problems characterized by their complexity, high dimensionality, hard nonlinearities and vague or imprecise knowledge. Most typical soft computing methods are ANNs, fuzzy systems and evolutionary computation. Many of these techniques exhibit complementary aspects and hence, they provide very

## 7.1. Combined Heat and Power Plant

often better performance when combined in a cooperative way as hybrid approach rather than acting exclusively (e.g. neuro-fuzzy systems, evolutionary fuzzy systems, or neuro-evolutionary systems). Due to those interesting properties, soft computing methods are widely used for modeling industrial processes [NSH<sup>+</sup>13, IHK12, ZA12]. Soft computing methods have also been successfully applied to cogeneration plants for analysis/diagnosis, optimization and control or prediction purposes (see [RVMB14, BSE<sup>+</sup>16] for a comprehensive literature review).

The cogeneration plant is located in Monzón (Huesca), in the north of Spain<sup>1</sup>. The main systems of the plant are: four internal combustion engines, four refrigeration engine circuits, an exhaust steam boiler, a steam turbine condenser, a steam turbine, and a slurry drying process. The plant produces electricity by means of the combustion engines and the steam turbine. The steam is generated with the heat of the exhaust gases of the four engines. Part of this heat is also used in a slurry drying process being the slurry provided by nearby farms.

The four internal combustion engines are all identical, i.e. with the same characteristics and the nominal power of each being 3700 kW. They are organized into two banks with eight cylinders each and the fuel used for the combustion is natural gas. The engines exchange heat with two circuits that use water from the cooling towers. A cooling circuit refrigerates the mixture of air and fuel mix around 50 °C and the other circuit preheats the intake air to around 35 °C. The engines generate electrical energy, which is sold, and flue gases. Each engine has a diverter, which sends the flue gases to an exhaust steam boiler when the engine is working above 50 % of rated power, or to the chimney if the rated power is below 50 %. Engines are usually above this threshold, and therefore the flue gases go to the exhaust steam boiler most of the time.

Next, the heat from the exhaust steam boiler is used by the steam generator to create steam at 400 °C and 22.5 bar. This steam feeds the steam turbine to generate more electricity, with 1000 kW of nominal power. The condenser of the steam turbine uses water from the cooling towers to condensate the steam from the steam turbine and recirculate it to the system. In addition, as in the engines, the power generated with the steam turbine is sold.

The slurry from the farms consists of approximately 6 % solids. Firstly, a mechanical treatment is carried out to remove the solid part from the rest using rotatory equipment. Then, a chemical treatment in the liquid part is performed to remove the chemical load. After that, the heat treatment uses the result of the chemical treatment to separate the condensables from non-condensables in an evaporator using superheated water generated in the exhaust steam boiler

---

<sup>1</sup><http://www.energyworks.com>. Accessed 13.12.2017

## 7. Real-world Applications

Table 7.1.: Nomenclature of parameters, variables and objectives used for the cogeneration optimization problem including physical quantities in which they are measured.

$Div_i$	Diverter Engine $i \in \{A, B, C, D\}$ (%)
$F_{Cond}$	Condensate effluent flow ( $\text{kg h}^{-1}$ )
$F_{Ev}$	Flow fed evaporator ( $\text{kg h}^{-1}$ )
$F_{FlueGas}$	Flue gases flow ( $\text{kg h}^{-1}$ )
$F_{Gas_i}$	Flow natural gas Engine $i \in \{A, B, C, D\}$ ( $\text{m}^3 \text{h}^{-1}$ )
$F_{Steam}$	Steam flow to steam turbine ( $\text{kg h}^{-1}$ )
$H_{Amb}$	Ambient humidity (%)
LHV	Low heating value ( $\text{kW h m}^{-3}$ )
$p_{Cond}$	Condenser pressure (bar)
$p_{Ev}$	Evaporator pressure (bar)
$p_{St\_Gen}$	Steam generator pressure (bar)
$P_i$	Rated power Engine $i \in \{A, B, C, D\}$ (%)
$P_{ST}$	Turbine power (kW)
$P$	Total generated power (kW)
$Q_{fuel}$	Used fuel (kW)
$T_{Bank_{ij}}$	Temperature gases Bank $i \in \{1, 2\}$ Engine $j \in \{A, B, C, D\}$ ( $^{\circ}\text{C}$ )
$T_{Amb}$	Ambient temperature ( $^{\circ}\text{C}$ )
$T_{H_2O\_Ex}$	Water temperature exchange ( $^{\circ}\text{C}$ )
$T_{H_2O\_SH}$	Water superheated temperature ( $^{\circ}\text{C}$ )
$T_{H_2O\_TH}$	Water temperature tubular heater ( $^{\circ}\text{C}$ )
$T_{H_2O\_Tow}$	Water temperature cooling tower ( $^{\circ}\text{C}$ )
$T_{in_{ij}}$	Temperature intake air Bank $i \in \{1, 2\}$ Engine $j \in \{A, B, C, D\}$ ( $^{\circ}\text{C}$ )
$T_{Mixt\_Eng_i}$	Water temperature to cooling the mixture Engine $i \in \{A, B, C, D\}$ ( $^{\circ}\text{C}$ )

(water with a temperature around  $120^{\circ}\text{C}$ ). A tubular heater is used to recirculate the effluent to the evaporator and preheat it. The tubular heater uses water from the refrigeration circuit, which preheats the intake air of the engines. The non-condensable part goes with the solid part resulting from the mechanical treatment and is sold as fertilizer. The condensable effluent is condensed again with the water from the cooling towers. Finally, the sterilizer uses the heat from the superheated water to purify the condensed effluent, thereby obtaining water suitable for irrigation. An overview of all relevant parameters, variables and objectives is given in Table 7.1.

As explained, the plant components are modeled by means of artificial ANNs. ANNs are a method from machine learning that imitate the behavior of the human brain in performing various computational tasks. A basic ANN consists of interconnected layers each of which possesses a certain number of neurons (Figure 7.1). The edges that connect the neurons of the different layers are characterized by weights that allow each neuron to prioritize its inputs. With the exception of the input neurons, the output of each neuron is obtained by summing the weighted inputs and subsequently applying an activation function to the resulting weighted sum. For a comprehensive introduction to ANNs see, e.g. [Ert11, Chp. 9].

## 7.1. Combined Heat and Power Plant

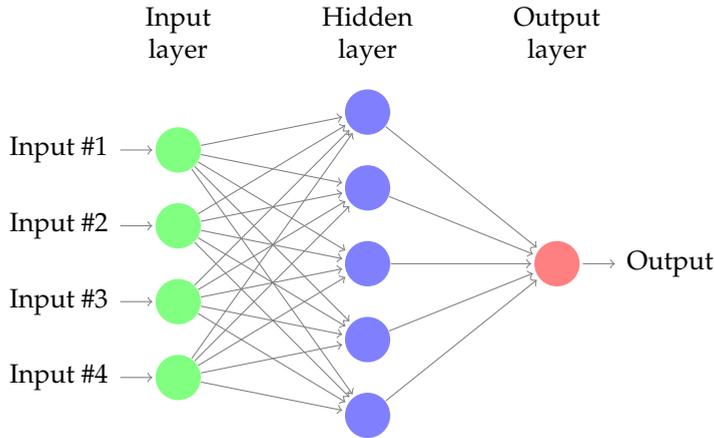


Figure 7.1.: Illustration of an ANN. Inputs are propagated from the input layer through hidden layers to compute outputs at the output layer. Each neuron computes an output by weighing and aggregating its inputs. Illustration by Kjell Magne Fauske [Fau].

Each plant component is represented by a single ANN. The inputs of the ANN are quantities measured during the operation of the plant that affect the components output. The steam turbine, for example, produces the power  $P_{ST}$ , which depends on the steam generator pressure  $p_{StGen}$ , the steam flow  $F_{Steam}$  and the condenser pressure  $p_{Cond}$ . The ANNs are linked (connected) by means of shared or common variables (e.g. an output variable of a network can be an input variable of one or several other networks). The global model is shown in Table 7.2 where a total of twelve ANNs are given: four for the engines, four for the engine cooling circuits, one for the exhaust steam boiler, one for the steam turbine condenser, one for the steam turbine and finally another one for the slurry drying process.

Multilayer perceptrons trained by the back-propagation algorithm were used as ANN model. A multilayer perceptron is a feedforward ANN – edges only point from neurons of previous to direct successor layers – that may possess multiple hidden layers. The back-propagation algorithm is a training algorithm that finds optimal weights for the network edges. Edge weights for a given set of input data are considered optimal if the error consisting of the difference between the output recorded in data set and the output generated by the ANN is minimal. The back-propagation algorithm feeds the error back to the ANN to iteratively update the weights. See [Ert11] for a detailed description of the back-propagation algorithm.

## 7. Real-world Applications

Table 7.2.: Plant components modeled by ANNs. Inputs marked in red represent decision variables used for the optimization of the plant.

Plant component	Inputs of the model	Output
Cooling Engine A	$T_{H2O\_Ex}, T_{H2O\_Tow}, P_A$	$T_{Mist\_Eng\_A}$
Cooling Engine B	$T_{H2O\_Ex}, T_{H2O\_Tow}, P_B$	$T_{Mist\_Eng\_B}$
Cooling Engine C	$T_{H2O\_Ex}, T_{H2O\_Tow}, P_C$	$T_{Mist\_Eng\_C}$
Cooling Engine D	$T_{H2O\_Ex}, T_{H2O\_Tow}, P_D$	$T_{Mist\_Eng\_D}$
Engine A	$T_{in\_1A}, T_{in\_2A}, T_{Amb}, H_{Amb}, LHV,$ $T_{Bank\_1A}, T_{Bank\_2A}, T_{Mist\_Eng\_A}, P_A, DIV_A$	$F_{Gas\_A}$
Engine B	$T_{in\_1B}, T_{in\_2B}, T_{Amb}, H_{Amb}, LHV,$ $T_{Bank\_1B}, T_{Bank\_2B}, T_{Mist\_Eng\_B}, P_B, DIV_B$	$F_{Gas\_B}$
Engine C	$T_{in\_1C}, T_{in\_2C}, T_{Amb}, H_{Amb}, LHV,$ $T_{Bank\_1C}, T_{Bank\_2C}, T_{Mist\_Eng\_C}, P_C, DIV_C$	$F_{Gas\_C}$
Engine D	$T_{in\_1D}, T_{in\_2D}, T_{Amb}, H_{Amb}, LHV,$ $T_{Bank\_1D}, T_{Bank\_2D}, T_{Mist\_Eng\_D}, P_D, DIV_D$	$F_{Gas\_D}$
Exh. steam boiler	$P_{StGen}, F_{FlueGas}$	$F_{Steam}$
Steam turbine	$T_{H2O\_Tow}, T_{ST\_Cond}$	$P_{Cond}$
Condenser		
Steam turbine	$P_{StGen}, F_{Steam}, P_{Cond}$	$P_{ST}$
Slurry process	$P_{EV}, T_{H2O\_SH}, T_{H2O\_Ex}, T_{H2O\_TH}, F_{Cond}$	$F_{Ev}$

Preliminary testing was conducted to determine a suitable topology for the multilayer perceptrons. It was concluded that the accuracy did not show any significant improvement after increasing the number of hidden layers. Therefore, the simplest option was selected: only one hidden layer. Similarly, some initial tests were made with different number of hidden nodes. It was concluded that when the number of neurons increases beyond twice the number of inputs (i.e., a common practical rule), results barely improve. Therefore, the adopted criterion in all the models is, using twice as many hidden nodes as the number of inputs.

To train and test the ANNs, a large data set was collected through a one-year observation process in the real plant. In total, 213 parameters were identified as being potentially relevant for training and validating the ANNs. Their values have been measured and retrieved with a resolution of one minute during the whole period of observation. Firstly, a careful analysis of the data was performed to choose the most relevant variables and also to filter outliers, missing data or uninformative variables. Next, based on previous knowledge of the system physics and also on a trial and error process, input variables for each ANN were determined. To make the huge data set more tractable, the data set resolution was changed to 10-minute separated values. This action can be realized, because it was observed that observed parameters change very little in that period, due to the slow dynamics of the plant. Thus, a total of about 40 000 samples were obtained for each variable. Now, for making the ANNs capable of modeling the different dynamics of the cogeneration process throughout the whole year, the data was partitioned into a training and test set: data from odd

## 7.1. Combined Heat and Power Plant

Table 7.3.: Mean absolute error (MAE) for training and testing samples for each neural network.

	Structure	Training MAE	Testing MAE
Cooling Engine A	3/6/1	0.21 %	0.23 %
Cooling Engine B	3/6/1	0.28 %	0.26 %
Cooling Engine C	3/6/1	0.10 %	0.13 %
Cooling Engine D	3/6/1	0.49 %	0.33 %
Engine A	10/20/1	0.39 %	0.42 %
Engine B	10/20/1	0.41 %	0.41 %
Engine C	10/20/1	0.38 %	0.42 %
Engine D	10/20/1	0.38 %	0.37 %
Recovery Boiler	2/4/1	0.61 %	0.63 %
Steam Condenser	2/4/1	1.01 %	0.96 %
Steam Turbine	3/6/1	0.67 %	0.70 %
Slurry Process	5/10/1	2.35 %	2.52 %

months (i.e. January, March, May, ...) are used to train the models and data from even months (i.e. February, April, June, ...) are used to test the modeling performance of the trained ANNs. The whole training/testing process has been carried out by using the *OPTIBAT trainer*. In addition, all the variables have been normalized.

The results of the modeling are presented in Table 7.3, where the mean absolute error  $MAE = \frac{1}{K} \sum_{i=1}^K |y_i - y'_i|$  between the desired ( $y_i$ ) and actual ( $y'_i$ ) output with  $K$  being the sample size, for both the training and the testing phase, is shown. The errors observed are very small at less than 1 %. Only the slurry process exhibits a slightly higher error, which can be attributed to little information about its behavior. Even in the testing case, where the ANNs deal with unseen data, the error is quite small. Note that the testing error represents the model's behavior better than the training error as it contains unseen data during the training of the models. Hence, the testing error is used to evaluate the model's predictive behavior. For all the models, the difference between the training error and the testing error was always less than 0.3 %. This means that the models were capable of learning the dynamic of the systems and can make accurate predictions when dealing with unseen data. These results validate the modeling performance of the trained ANNs.

Once the CHP plant has been modeled by means of the connected ANNs, the next step is to carry out an optimization process to improve the efficiency of the plant operation. In particular, a focus is put on three performance objectives: 1) minimizing the amount of used fuel  $Q_{fuel}$  (i.e. natural gas flow); 2) maximizing the useful thermal energy  $F_{Ev}$  (i.e. flow of the fluent in the evaporator) and 3) maximizing the generated power  $P$ . To carry out the optimization, a total of twelve real-valued decision variables are available in the plant; that is, a set of input variables whose values can be changed freely (within certain

## 7. Real-world Applications

bounds) by the plant operator. These twelve variables are those highlighted in red in Table 7.2. The mathematical formulation of the corresponding MOOP is as follows:

$$\min_{T_{in_{ij}}, p_{St\_Gen}, p_{Ev}, T_{H2O\_SH}} (Q_{fuel}, -F_{Ev}, -P) \quad \text{s.t.} \quad (7.1a)$$

$$F_{Gas\_A} + F_{Gas\_B} + F_{Gas\_C} + F_{Gas\_D} = Q_{fuel} \quad (7.1b)$$

$$P_A + P_B + P_C + P_D + P_{ST} = P \quad (7.1c)$$

$$30^\circ\text{C} \leq T_{in_{ij}} \leq 38^\circ\text{C} \quad \forall i \in \{1, 2\}, j \in \{A, B, C, D\} \quad (7.1d)$$

$$20 \text{ bar} \leq p_{St\_Gen} \leq 22 \text{ bar} \quad (7.1e)$$

$$0.13 \text{ bar} \leq p_{Ev} \leq 0.17 \text{ bar} \quad (7.1f)$$

$$110^\circ\text{C} \leq T_{H2O\_SH} \leq 125^\circ\text{C} \quad (7.1g)$$

### 7.1.2. Optimization of a Complex Cogeneration Process

The combined approach of using ANNs as black box functions may be applied in conjunction with any optimization algorithm that is able to handle real-valued decision variables. For this reason, several state-of-the-art, MOEAs, which use different search strategies, are considered for solving the proposed optimization problem. The first step of the optimization procedure consists of computing a finite set of points representation of the Pareto front (Section 4.1.2) to attain a deeper understanding of the problem structure and to study the tradeoffs that occur between the different objectives. The same algorithms and their configuration that are used to benchmark ESPEA in Section 5.3.2 are retained to approximate the Pareto front of the cogeneration optimization problem: ESPEA (Algorithm 1), IBEA (Algorithm 6), MOEAD (Algorithm 9), NSGA-II (Algorithm 4), NSGA-III (Algorithm 10), SMPSO (Algorithm 7) and SPEA2 (Algorithm 5).<sup>2</sup>

As explained before, the performance of the CHP is influenced by 213 different parameters of which 36 were found to have a significant impact as indicated in Table 7.2. While twelve of these parameters may be manipulated by the plant operator as decision variables, there still exist 24 parameters, whose different combinations of values potentially affect the optimization effort but cannot be manipulated. For this reason 39 parameter observations were randomly picked from the database, which come from a week in February and serve as representative sample. For each observation, the values of those 24 parameters were

<sup>2</sup>AbYSS [NLA<sup>+</sup>08] instead of SPEA2 was considered in the computational study in [BSE<sup>+</sup>16]. In order to retain a consistent selection of algorithms throughout this work, AbYSS was replaced by SPEA2. AbYSS was not among the top performing algorithms in [BSE<sup>+</sup>16] and SPEA2 is the top performing algorithm in Section 5.3.2 making the exchange reasonable.

Table 7.4.: Cogeneration study – IGD. Mean and standard deviation (as subscript) of median IGD across all problem instances. Best performance is highlighted in green, second best performance in blue.

ESPEA	IBEA	MOEAD	NSGA-II
1.72e-4 <sub>1.89e-5</sub>	1.45e-3 <sub>1.83e-4</sub>	1.44e-3 <sub>1.66e-4</sub>	3.63e-4 <sub>2.09e-5</sub>
NSGA-III	SMPSO	SMS-EMOA	SPEA2
1.18e-3 <sub>1.26e-4</sub>	3.39e-4 <sub>1.97e-5</sub>	1.46e-3 <sub>1.61e-4</sub>	2.35e-4 <sub>2.05e-5</sub>

extracted. The 24 parameters of each of the 39 observations serve as individual problem instances for the computational study. The objective of this study is to identify the algorithm that delivers the best performance by choosing optimal values for the twelve decision variables across all 39 problem instances. Each algorithm was run 100 times on every problem instance employing a population size of 100. The algorithms used the same configuration as for the study in Section 5.3.2 (Table E.2). 50 000 function evaluations were performed per run. Preliminary tests have revealed that the populations of the algorithms assessed in this study become evolutionary stable at 50 000 evaluations, implying that the population exhibits little to no movement in the objective space after said number of function evaluations have passed.

IGD (Definition 57) was chosen as performance metric, since it captures both convergence and diversity. Since the Pareto fronts of the problem instances are unknown, all nondominated solutions obtained across all algorithm runs of a single problem instance are utilized as reference front (an example is given in Figure 7.2). Objective values were normalized to mitigate the effect of different scales. The study was performed within the jMetal framework version 4.5 [DN11] and the code is available online [Bra].

A preliminary analysis has revealed that the performance of an individual algorithm only differs marginally across the different problem instances. This observation indicates that the proposed approach is very robust with respect to the parameters that cannot be influenced by the operator. For the sake of clarity, only a summary of the results across all problem instances is provided in Table 7.4. Full results are provided in the appendix in Table G.13.

The study results demonstrate that there exist clear performance differences between individual algorithms. Values of the IGD metric differ by a factor of ten from best to worst. This implies that the choice of algorithm greatly influences the optimization outcome. Best results are obtained using ESPEA, whereas NSGA-II, SMPSO and SPEA2 show also good performances. IBEA, MOEAD,

## 7. Real-world Applications

NSGA-III and SMS-EMOA, on the other hand, trail behind. The smallest average IGD is achieved by ESPEA.

A statistical analysis was performed to check the results for statistical significance. The same procedure that was applied in Sections 5.3.2 and 6.3.2 is used for this purpose. The Kruskal-Wallis test confirmed significant performance differences on all 39 problem instances (Table G.19). ESPEA was found to outperform all other algorithms with confidence at a 95 % level on each problem instance (Table G.20). Therefore, ESPEA should be considered the supreme choice in obtaining finite set of points approximations of the Pareto front of the cogeneration problem.

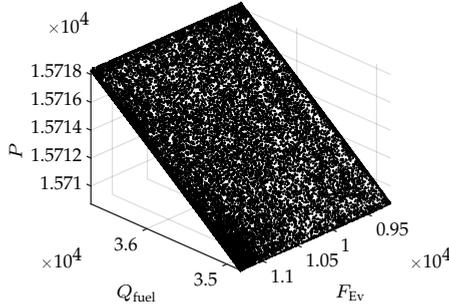


Figure 7.2.: Pareto front of problem instance 2 out of 39 of the cogeneration optimization problem. The front is a collection of all non-dominated solutions that were retrieved in final populations during the study. The Pareto front that these points describe can be approximated by a plane:  $P = 15530 + 5.13e-3F_{Ev} - 6.76e-7Q_{fuel}$ .

A closer analysis of the Pareto front reveals possible explanations for the performance differences observed. Figure 7.2 shows the Pareto front of a cogeneration optimization problem instance. The rectangular shape of the front suggests that all Pareto optimal points lie on a plane. A regression analysis has indeed confirmed that, for 38 out of 39 problem instances, all points can be fitted in a plane with a coefficient of determination of one and a root mean square error of about 0.019.<sup>3</sup> Interestingly, the Pareto front is almost a linear function, although the problem itself is not. As possible explanation for this observation may be traced to the activation function used by the neural network. A logistic function  $1/(1 + \exp(-x))$  was used, which can be approximated by a piecewise linear curve.

<sup>3</sup>The Pareto front of problem instance 1 out of 39 (referenced as CG0 in Table G.13) consists of two different planes.

## 7.1. Combined Heat and Power Plant

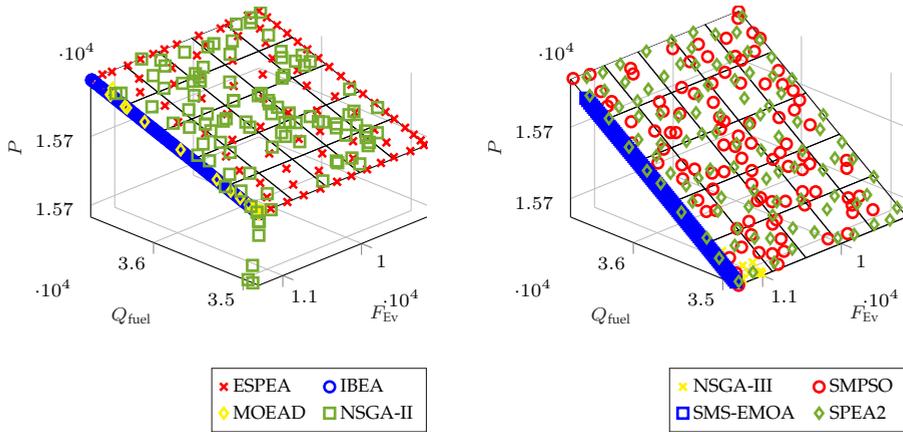


Figure 7.3.: Exemplary search results of ESPEA, IBEA, MOEAD, NSGA-II, NSGA-III, SMPSO, SMS-EMOA and SPEA2 on problem instance 2 of the cogeneration optimization problem for illustrating their performance.

Figure 7.3 offers an explanation to why algorithm performances can be divided into two tiers. ESPEA, NSGA-II, SMPSO and SPEA2 capture the extent of the Pareto front in its entirety, whereby ESPEA achieves the most equidistant approximation. NSGA-II obtains several dominated points as indicated in the plot. IBEA, MOEAD, NSGA-III and SMS-EMOA focus mainly on a single edge of the front. It is reasonable to assume that applying MOEAD and NSGA-III with reference points from [DD98] as it is suggested in [LZ09, DJ14] is problematic given the presented front. The reference points presumably do not cover the front equally, which leads to a strong focus on boundary solutions. A better choice of reference points might ameliorate this issue. Hypervolume-based methods, such as SMS-EMOA and the IBEA configuration used in this study, seem to struggle with the geometry of the front. One may speculate that, despite objective normalization, boundary points yield the highest hypervolume contributions on plane-shaped fronts. Recent studies have revealed that the success of hypervolume-based MOEAs highly depend on the geometry of the front [IISN17]. Although the figures only depict single runs, these basic observations could be confirmed for other problem instances and different runs as well.

Robustness is another aspect of algorithm performance that is of interest in the current setting. In practice, it is usually not feasible to conduct 100 runs and choose the most preferable option from this pool of alternatives. If there is little variability in the composition of the Pareto front approximation between individual runs, however, it may be concluded that every single run yields a

## 7. Real-world Applications

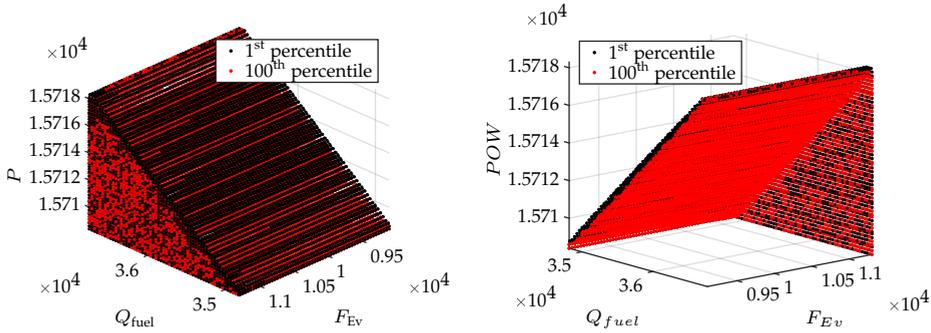


Figure 7.4.: Surface attainment plot of ESPEA for the 1<sup>st</sup> and 100<sup>th</sup> percentile of problem instance 2 from the front (left) and the back (right). The 1<sup>st</sup> percentile depicts the space that is dominated by all ESPEA runs combined, while the 100<sup>th</sup> percentile shows the space that is dominated in all of the 100 runs.

satisfying approximation of the Pareto front. One way of assessing robustness is considering the surface attainment [FF96] across multiple runs. Surface attainment measures the space that is dominated by a given approximation set. When measured across multiple runs, surface attainment yields the space that is dominated in a given percentage of runs. Figure 7.4 provides an example for the visualization of the surface attainment of ESPEA of the 1<sup>st</sup> and 100<sup>th</sup> percentile. Both surfaces nearly coincide, suggesting that the optimization approach is very robust across different runs. Similar observations were made for other problem instances.

The multiobjective approach generates a set of candidate solutions among which a DM chooses an option that fits his preferences best. In the present context, there exists an measure that may be used to evaluate the efficiency of the cogeneration process. In this work, the fuel efficiency is defined as the quotient of the total power generated by the unused energy contained in the fuel:

$$\varepsilon_{EE} = \frac{100 \cdot P}{Q_{\text{fuel}} - F_{\text{Ev}}/0.9}. \quad (7.2)$$

A question that needs to be addressed in this context is, whether the multiobjective approach is suited to find a solution that maximizes the efficiency of the cogeneration process. Table 7.5 shows the average of the best fuel efficiency achieved by each algorithm. Detailed results are contained in the appendix in Table G.14. The figures reveal that ESPEA, IBEA, MOEAD and SMP SO obtain roughly the same fuel efficiency. The fuel efficiency of SMS-EMOA is

Table 7.5.: Cogeneration study – fuel efficiency. Mean and standard deviation (as subscript) of median fuel efficiency across all problem instances. Best performance is highlighted in green, second best performance in blue.

ESPEA	IBEA	MOEAD	NSGA-II
70.5597 <sub>7.8805e-4</sub>	70.5588 <sub>4.5023e-4</sub>	70.5605 <sub>4.3261e-4</sub>	70.4152 <sub>0.0154</sub>
NSGA-III	SMPSO	SMS-EMOA	SPEA2
70.4846 <sub>0.0067</sub>	70.5605 <sub>4.3270e-4</sub>	70.5394 <sub>0.0436</sub>	70.2120 <sub>0.0159</sub>

marginally worse. Larger performance differences are observed for NSGA-II, NSGA-III and SPEA2. Overall, the results show that each algorithm is able to find fuel-efficient solutions. The average fuel efficiencies obtained by the algorithms are too similar to warrant a statistical analysis, since the expected effect size is negligible.

If a choice rule such as (7.2) is given, it makes sense to focus the search from the beginning on those regions of the Pareto front that yield the highest efficiency. In a multi-objective context, however, a DM is usually not only interested in obtaining a preconceived optimum, but also in comparing his choice to other options available [Roy96, KT79]. Additionally, situations may occur in which a deviation from a fuel-efficient operation is prudent. Short-term fluctuations in the electricity prices might encourage the plant operator to increase power output at the cost of increasing fuel consumption, thus generating larger profits. Changes in ambient temperature possibly affect the need for heat during the slurry drying process. Differences in the quantity and cost of slurry supply might have an effect on the necessity to conserve or spend fuel.

ESPEA's charge mechanism enables the plant operator to obtain more operating schedules that are close to but differ from the fuel efficiency optimum to accommodate the needs for deviation outlined in the previous paragraph. Figure 7.5 illustrates the effect of using the fuel efficiency as scalarized preference information in ESPEA. Equation (7.2) was additionally raised by the power of three to give the search an even stronger focus to obtain fuel efficient schedules. The density of solutions is higher in those regions that exhibit a high fuel efficiency. At the same time, an approximation to the Pareto front in its entirety is retained enabling the DM to compare the most efficient schedule to other alternatives available.

Applying ESPEA's charge mechanism focuses the search on generating more solutions that are close to the efficiency optimum. In this context, it appears reasonable to compare ESPEA to a SOO algorithm with respect to the efficiency

## 7. Real-world Applications

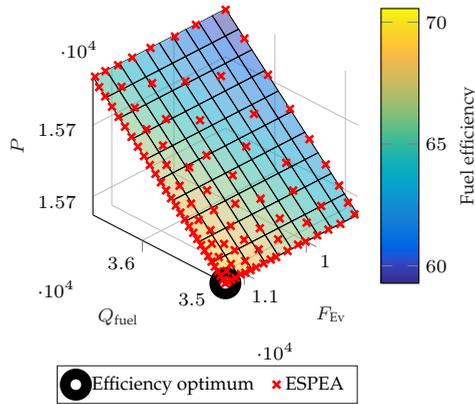


Figure 7.5.: A single run of ESPEA using fuel efficiency as scalarization function on problem instance 31 of the cogeneration optimization problem.

optima that both algorithms obtain. To allow a fair basis of comparison, a GA (Algorithm 15) was chosen that uses the same evolutionary operators as ESPEA and population size of 100. The GA uses SBX and polynomial mutation. Both operators are configured in the same way as in ESPEA. Figure 7.6 depicts a comparison of both algorithms' performances across all test problems. Although the GA converges faster, especially within the first 150 iterations, the performances of both algorithms align the more function evaluations are performed. Considering absolute values, the performance differences are almost negligible.

Since the cogeneration optimization problem as presented in this section is embedded in a real-time application, algorithm run times are a critical issue in this context. As mentioned before, the thermodynamical processes inside the plant change rather slowly. Therefore, the plant is expected to be reconfigured in regular intervals of 15 minutes. After 15 minutes have elapsed, the optimization is performed using the values of the 24 significant parameters that are currently observed. The plant rescheduling is performed automatically by the algorithm choosing the solution of the final population possessing the highest efficiency. A plant operator may optionally review the algorithm's choice by analyzing the tradeoffs observed on the Pareto front. A single algorithm run should therefore take considerably less than 10 minutes on commercial off-the-shelf medium-class computer hardware. With the exception of ESPEA and SMS-EMOA, all algorithms finish 50 000 function evaluations within less than ten seconds on an *Intel Core i5-4300U* processor with 8GB RAM running *Microsoft Windows 8.1*. A single ESPEA run takes about half a minute, whereas an SMS-EMOA execu-

## 7.2. Building Energy Management Systems

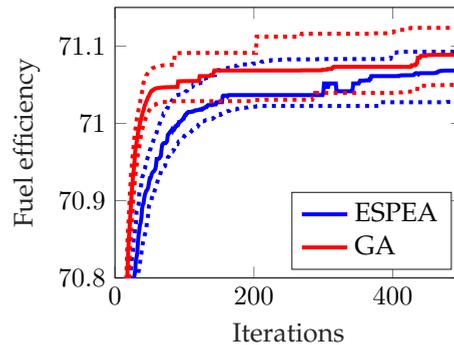


Figure 7.6.: Comparison of ESPEA using fuel efficiency as scalarization function to a GA. The figure shows the median (solid line) and the 95<sup>th</sup> percentile (dotted line) of the best efficiency observed in each iteration and for every problem across all 100 runs for each algorithm. The median of medians and the 95<sup>th</sup> percentile of 95<sup>th</sup> percentiles across all 39 problems are depicted in the figure.

tion can take up to ten minutes. Although run-times also highly depend on the implementation and the chosen programming language, the most important factor is the computational complexity of executing a single iteration of a given optimization algorithm. ESPEA and SMS-EMOA are steady-state algorithms implying that more costly operations are performed per function evaluation. As elaborated on in Section 4.1.2, SMS-EMOA computes the hypervolume contribution (Figure 4.2a) of each population member in every iteration, which is an NP-hard problem [BF09] resulting in a very high run-time. Considering all this, one may draw the conclusion that every algorithm assessed in this study is suitable for an on-line implementation with the exception of SMS-EMOA.

## 7.2. Building Energy Management Systems

*Declaration: Parts of this section have been published in [BDMS16]. Some passages are quoted verbatim.*

BEMSs allow the automated operation of appliances and other devices in commercial and residential buildings. Automated operation may pursue multiple, potentially conflicting goals. This section presents a framework for modeling devices in a residential building and finding optimal schedules for their operation. Three conflicting objectives are considered: 1) minimizing costs of total energy consumption, 2) minimizing carbon dioxide emissions caused by energy consumption and 3) minimizing thermal discomfort of the residents. In

## 7. Real-world Applications

the presence of multiple, conflicting objectives, a DM is required to choose an operating schedule that suits her preferences. Choosing an optimal operation schedule among a large set of Pareto optimal solutions, however, is arduous. On the other hand, applying a priori technique methods to obtain a single optimal schedule is difficult if the DM has no prior knowledge about the possible tradeoffs between the different objectives, on which she can build her preferences. Local scalarization optima may bridge the gap between these two approaches in providing the DM only with a limited set of interesting schedules based on a scalarization function that is a rough estimate of the DM's true preferences.

### 7.2.1. A Multi-objective Framework for Building Energy Management Systems

Modern buildings possess devices and systems whose operating times and modes can be controlled by automated BEMSs [MMAS16]. The scope of such systems is finding optimal operating schedules for these devices that adhere to constraints and preferences set by the residents. Devices may be categorized according to the degree of control the BEMS can exercise over them (Table 7.6). Non-optimizable devices are controlled by the residents. The BEMS is not allowed to make any autonomous decisions about how and when non-optimizable devices are operated. Such devices include white goods such as stoves and ovens or components of home entertainment systems such as TV sets, video game consoles or personal computers.

There also exist devices over which neither the residents nor the BEMS exercise control. Such devices are also classified as non-optimizable and include, for example, photovoltaic (PV) systems. PV systems are installed on building rooftops and transform solar radiation into electricity by using the photovoltaic effect. Electricity generated by the PV system can either be consumed in the building or fed into to the electric grid. The amount of electricity generated depends mostly on the intensity of the solar irradiance and cannot be changed by an operating mechanism.

Deferable appliances are devices, whose starting times may be delayed and thus controlled by the BEMS. A resident usually specifies a temporal degree of freedom, which states the maximum time the device may be delayed and thereby implies a time windows within which the device has to be operated. A dishwasher is an example for a deferrable appliance. Residents usually do not require a dishwasher to start right away once it is has been filled. Instead, residents rather specify the dishwasher to be finished by a certain time. For

## 7.2. Building Energy Management Systems

Table 7.6.: Overview of device types in residential buildings. See [BDMS16].

Load-flexible	Inter-ruptible	Description	Examples
		Device may only be controlled by the user	Stove, oven, TV, video game consoles
✗	(✓)	Starting time of the device may be optimized within a defined period	Conventional dishwasher, washing machine
✗	✓	Operation cycle of device may be split into one or more phases that are separated by pauses	Interruptible dishwasher, micro-CHP plant
✓	✗	Device has alternative load profiles for the same operation cycle	Lighting, heat pump, heating
✓	✓	Operation cycle consists of multiple phases for each of which alternative profile may be chosen	Air-conditioning, gas-fired boiler, battery

example, a resident may load and program the dishwasher in the evening after dinner and requires dishes to be cleaned by the next morning.

Interruptible devices are appliances, whose operation cycle can be split into multiple phases. The execution of such devices may be paused between individual phases, while individual phases are non-interruptible. The general operation cycle of a dishwasher, for example, consists of the following phases: heating water, cleaning (rinsing), draining and drying. Modern dishwashers may perform multiple cycles of heating, cleaning and draining during a single execution. A dishwasher may be interrupted between some of these phases within their temporal degree of freedom. Of course, technical constraints limit the amount of time that is allowed to pass between phases. For instance, heated water eventually cools down rendering it less efficient for cleaning dishes. Any deferrable device can be modeled as an interruptible device possessing exactly one non-interruptible phase. Deferrable devices are therefore listed in Table 7.6 as interruptible in parenthesis.

There also exist devices that possess phases of variable length and a variable

## 7. Real-world Applications

number of phases. For example, a micro-CHP plant in a residential building burns natural gas to generate electricity and heat, which may be consumed by HVAC devices. The BEMS may choose the starting and execution times of the micro-CHP plant as well as the number of start-ups. Each execution time can be understood as a phase.

Load-flexible devices possess alternative load profiles for the same operation cycle. A load profile defines the power consumption of a device across time during its operation. Heating elements are characterized, for example, as load-flexible. Radiators usually possess multiple heating levels, which result in different amounts of heat being emitted by the radiator. The BEMS may exercise control over such devices, for example, by temporarily lowering the heating level in favor of saving energy costs during periods of high electricity prices.

Finally, there exist devices that are both load-flexible and interruptible. Such devices may possess multiple load profiles for the same phase. A dishwasher, for example, may theoretically heat water at different heat levels. Lower levels require less energy but more time until the water has attained the necessary temperature for cleaning, whereas higher levels require more energy and less time. Devices of variable phase length and variable number of phases may also be load-flexible. Air-conditioning, for example, can be run at different cooling levels and temporarily paused if necessary. Lowering the cooling level or completely turning off air-conditioning in times of peak electricity prices may save energy consumption costs.

Load-flexible, interruptible devices build the basis of the proposed modeling approach as any optimizable (or schedulable) device can be modeled as a load-flexible, interruptible device. A non-load-flexible, interruptible device is a load-flexible, interruptible device that possesses only a single load-profile for each phase. Equivalently, load-flexible, non-interruptible devices are load-flexible and interruptible devices whose operation cycle consists of a single phase.

The modeling of the optimization problem requires that each aspect of control, exercised by the BEMS on the devices in the building, is encoded into a decision variable. The BEMS optimizes the scheduling of devices within a specified time frame – the optimization horizon. For scheduling the devices, time is discretized into time slots of fixed lengths. The variable  $t$  denotes such a time slot. If a device is scheduled to start in  $t$ , it is executed at the beginning of  $t$ . Each device  $j$  possesses an earliest starting time, which is called release time  $r_j$  and a latest finishing time denoted by deadline  $d_j$ . Both release time and deadline are provided by the residents of the building. The length of an individual phase  $k$  is denoted by  $p_{j,k}$ . Interruptibility is modeled by associating every phase with a decision variable  $s_{j,k}$ . Each  $s_{j,k}$  is an integer that states the delay, with which phase  $k$  of device  $j$  is executed with respect to the earliest time it may be executed. Hence,  $s_{j,k} = 0$  implies that phase  $k$  is directly executed after phase

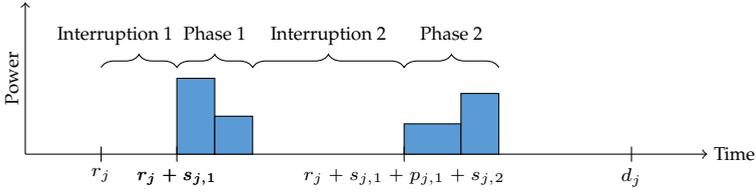


Figure 7.7.: Illustration of the encoding of a two-time interruptible device. Adapted from [BDMS16].

$k - 1$ . The vector  $\mathbf{s}^j := (s_{j,1}, s_{j,2}, \dots)$  contains every delay of each individual phase. If device  $j$  is deferrable, then  $\mathbf{s}^j$  becomes a vector of length one, i.e. a scalar. See Figure 7.7 for an illustration of an interruptible device. Constraint-handling is simplified using the proposed encoding if a heuristic is applied, since successive phases are always executed consecutively. Mathematical programming formulations, however, require a different modeling approach (see e.g. [SWSJ11]).

Operation modes can also be represented using an integer encoding. For each device  $j$  at phase  $k$  operation mode  $a_{j,k,l}$  is chosen, where  $\mathbf{a}^{j,k}$  is a vector of feasible operation modes for device  $j$  in phase  $k$ . An operation mode  $a_{j,k,l}$  is represented by an integer where  $a_{j,k,l} = l$ , i.e. operation mode  $l$  of phase  $k$  and device  $j$  is encoded by integer  $l$ . If device  $j$  possesses only a single operation mode in phase  $k$  then  $a_{j,k,l} = a_{j,k,1} = 1$ , i.e.  $l = 1$ . In the implementation of the simulation, decision variables for the operation modes are only introduced if there is more than one operation mode to choose from.

HVAC devices can be started an arbitrary number of times by the BEMS during the optimization horizon. To model this aspect in the problem formulation, a decision variable  $n^j$  is introduced that expresses the number of phases. Consequently,  $\mathbf{s}^j$  is a vector of length  $n_j$ . As indicated before, BEMS may also dictate the execution time of HVAC devices. Thereby, the length  $p_{j,k}$  of phase  $k$  of any HVAC device  $j$  becomes a decision variable itself and is no longer a parameter of the inherent technical design of the device. For practical reasons, the number of potential start-ups is capped in the simulation to a maximal number  $n^{\max}$ . The genome reserves space for  $n^{\max}$  delays, phase lengths and operation modes for each HVAC device. If for any  $j$  it holds that  $n_j < n^{\max}$ , the remaining  $n^{\max} - n_j$  entries in the genome are simply ignored.

Summarizing, each optimizable device is characterized by four decision variables of which three are vectors. These four variables define the schedule of the device.

## 7. Real-world Applications

- $\mathbf{s}^j$ : vector of delays for each individual phase
- $\mathbf{p}^j$ : vector of lengths of each individual phase
- $\mathbf{a}^j$ : vector of operation modes for each phase
- $n_j$ : number of operation cycles

As noted before, if any decision variable is not applicable to a given device, e.g. the phase lengths of a dishwasher are subject to the chosen operation mode and cannot be changed by the BEMS. They are simply ignored for this specific device instance.

The schedules of the devices in a building allow inferring their internal states at any time slot  $t$ . The internal states of the devices determine their current consumption of energy and provision of energy and services such as temperature regulation, which in turn affects the minimization of the objectives total energy costs, carbon dioxide emissions and thermal discomfort. The next step of the modeling consists of quantifying these internal states such that the resulting values can be used to compute objective values.

The first aspect that is considered in this context is the electricity consumption. For being able to perform the optimization, the BEMS is required to know the electric energy that every device  $j$  consumes during any time slot  $t$ . Let  $E_{j,k,l}$  denote the load profile of device  $j$  in phase  $k$  of operation mode  $l$ . The amount of energy a device consumes in a time slot  $t$  is equivalent to the integral of the load profile in  $t$ . Based on the chosen delay  $s_{j,k}$  and operation mode  $a_{j,k,l}$ , the electric energy that appliance  $j$  consumes in time slot  $t$  is expressed by the function  $E_{j,k,l}(t^{\text{ref}})$ . The time (i.e. the number of time slots) that has elapsed, since phase  $k$  has been started  $t^{\text{ref}}$  is defined relative to  $t$ :

$$t^{\text{ref}} = r_j + \left( \sum_{i=1}^k s_{j,i} \right) + \left( \sum_{i=1}^{k-1} p_{j,i} \right) - t. \quad (7.3)$$

The definition of the reference time  $t^{\text{ref}}$  enables the BEMS to determine how much energy device  $j$  consumes in time slot  $t$  given its choice of  $\mathbf{s}^j$ ,  $\mathbf{a}^j$ . The electric energy  $E_j(t)$  that device  $j$  consumes in time slot  $t$  can then be defined in the following way:

$$E_j(t) = \begin{cases} E_{j,k,l}(t^{\text{ref}}) & \text{if } t \in \left[ r_j + \left( \sum_{i=1}^k s_{j,i} \right) + \left( \sum_{i=1}^{k-1} p_{j,i} \right), r_j + \sum_{i=1}^k s_{j,i} + p_{j,i} \right] \\ E_j^{\text{standby}} & \text{else.} \end{cases} \quad (7.4)$$

The range definition of  $t$  for which the case  $E_{j,k,l}(t^{\text{ref}})$  applies coincides with the time slots in which phase  $k$  is active. Device  $j$  only consumes the energy

## 7.2. Building Energy Management Systems

$E_{j,k,l}(t^{\text{ref}})$  in  $t^{\text{ref}}$  if phase  $k$  is active. If no phase is active in  $t$  the device  $j$  consumes the standby consumption  $E_j^{\text{standby}}$ . Note that the standby consumption may vary in real-world appliances depending on which phase has been most recently finished, e.g. a dishwasher that has just heated the water for cleaning might maintain the water temperature by consuming more energy as long as the next washing phase has not started. For clarity and simplicity, this behavior is not explicitly considered in the formal model.

Equation (7.4) can also be used to model the provision of electricity within the building by schedulable devices. The function  $E_{j,k,l}(t^{\text{ref}})$  then yields the amount of energy device  $j$  produces in phase  $k$  given operation mode  $l$ . The value  $E_{j,k,l}(t^{\text{ref}})$  is negative if  $j$  generates electricity to distinguish between consumption and production. The micro-CHP plant, for example, burns natural gas to generate heat and electricity. In contrast to household appliances that possess fixed phase lengths, the BEMS can control the length of individual phases of phase-length-variable devices such as the micro-CHP plant. The load profile of any such device is then only considered within the confines of the phase length. A micro-CHP plant, for example, requires a start-up time, in which it produces less energy and heat, before it is fully operational and generates a constant power output once it has reached its optimal operating point. Note that the same load profile is used across all executed phases and  $E_j^{\text{standby}} = 0$  for any electricity-producing device.

In addition to appliances that consume electricity, there also exist devices that consume natural gas such as the micro-CHP plant. Equation (7.4) can also be used to model gas consumption. Instead of a load profile  $E_{j,k,l}$  a gas consumption profile  $G_{j,k,l}$  is associated with each gas consuming device and the function  $G_{j,k,l}(t^{\text{ref}})$  yields the gas consumption with respect to the time that has elapsed, since the device has been started. The gas consumption  $G_j(t)$  of a gas consuming device  $j$  can then be described by

$$G_j(t) = \begin{cases} G_{j,k,l}(t^{\text{ref}}) & \text{if } t \in \left[ r_j + \left( \sum_{i=1}^k s_{j,i} \right) + \left( \sum_{i=1}^{k-1} p_{j,i} \right), r_j + \sum_{i=1}^k s_{j,i} + p_{j,i} \right] \\ 0 & \text{else.} \end{cases} \quad (7.5)$$

Gas consuming devices do not burn any gas if they are not active and therefore have no standby consumption. Similar to the load profile of the micro-CHP plant, the gas profile  $G_{j,k,l}$  can also be used to model time-variable consumption in the start-up phase of gas consuming devices. Additionally, gas-consuming devices usually require a minimum downtime before they may be restarted. Such downtimes are included in the phase lengths, i.e. the phase

## 7. Real-world Applications

length  $p_{j,k}$  covers the period in which gas is burned and electricity is generated and the minimum downtime. A phase is therefore at least as long as the minimum downtime.

The final step of the device modeling is concerned with the provision of heat and cooling for regulating the temperature inside the building. HVAC devices consume either gas or electricity to generate heat or cooling. Air-conditioning uses electricity to cool down the temperature inside the building. The micro-CHP plant and the condensing boiler, on the other hand, first feed their generated heat into a hot water storage tank. Energy, in the form of warm water, is extracted from the hot water storage tank to heat the building. Residents also utilize hot water from the hot water storage tank for other purposes in the household, for example for cooking or showering.

The provision of heat by HVAC devices can be described in the same manner as the consumption of electricity and gas (Equations (7.4) and (7.5)). Let  $Q_{j,k,l}$  denote the energy output – either heat or cooling – of device  $j$  in phase  $k$  and operation mode  $l$ . The function  $Q_{j,k,l}(t^{\text{ref}})$  yields the current energy provision with respect to the time that has elapsed since phase  $k$  has been started:

$$Q_j(t) = \begin{cases} Q_{j,k,l}(t^{\text{ref}}) & \text{if } t \in \left[ r_j + \left( \sum_{i=1}^k s_{j,i} \right) + \left( \sum_{i=1}^{k-1} p_{j,i} \right), r_j + \sum_{i=1}^k s_{j,i} + p_{j,i} \right] \\ 0 & \text{else.} \end{cases} \quad (7.6)$$

The final component of the model of the residential building consists of the temperature model. A simplified model is chosen that considers only the energy flows between the building components. The temperature  $T(t)$  inside the building in any time slot  $t$  depends on the temperature in the previous iteration  $T(t-1)$  and the energy delta  $Q^\Delta(t)$  that is either added to or removed from the air inside the building in the form heat or cooling:

$$T(t) = T(t-1) + \frac{Q^\Delta(t)}{A \cdot h \cdot \rho^{\text{air}} \cdot c^{\text{air}}}, \quad (7.7)$$

where  $A$  and  $h$  are the area and height of the building, and  $\rho^{\text{air}}$  and  $c^{\text{air}}$  are the density and the specific heat capacity of air. The energy delta can be decomposed into heat added by the heating  $Q^{\text{heat}}(t)$ , losses or gains  $Q^{\text{loss}}(t)$  that occur as a result of the temperature difference between inside and outside the building, cooling caused by the air-conditioning  $Q_{\text{AC}}(t)$  and other sources  $Q^{\text{other}}(t)$  that include waste heat of household devices and residents present in the building:

$$Q^\Delta(t) = Q^{\text{heat}}(t) + Q^{\text{loss}}(t) - Q_{\text{AC}}(t) + Q^{\text{other}}(t). \quad (7.8)$$

## 7.2. Building Energy Management Systems

The cooling of the air-conditioning enters Equation (7.8) as negative term, since cooling removes energy in form of heat from the building. A subscript is used to index the cooling, since  $Q_{AC}(t)$  corresponds to the energy produced by the air-conditioning according to Equation (7.6). In contrast,  $Q^{\text{heat}}(t)$  is energy extracted from the hot water storage tank. The heat loss or gain of the building  $Q^{\text{loss}}(t)$  is computed as follows:

$$Q^{\text{loss}}(t) = (T^{\text{out}}(t-1) - T(t-1)) \cdot A^{\text{env}} \cdot h^{\text{loss}}, \quad (7.9)$$

where  $T^{\text{out}}$  is the outside temperature,  $A^{\text{env}}$  the area of the building envelope and  $h^{\text{loss}}$  is a heat transfer coefficient.

As mentioned before,  $Q^{\text{heat}}(t)$  is energy that is extracted from the hot water storage tank. In the proposed scenario, the extracted  $Q^{\text{heat}}(t)$  is managed by a controller and the BEMS does exercise direct influence on the amount of water that is extracted. Instead, the residents set a target temperature  $T^{\text{set}}$  for the building and the BEMS schedules the micro-CHP-plant and the condensing boiler such that the target is met. The necessary heat  $Q^{\text{NH}}(t)$  for maintaining the target temperature is calculated using the following equation:

$$Q^{\text{NH}}(t) = Q^{\text{loss}}(t) + (T^{\text{set}} - T(t-1)) \cdot A \cdot h \cdot \rho^{\text{air}} \cdot c^{\text{air}}. \quad (7.10)$$

The amount of heat that can be extracted from the hot water storage is limited by the temperature of the hot water storage tank. The hot water storage tank possesses a minimum temperature  $T_{\text{WS}}^{\text{heat}}$  above which hot water may be extracted for heating. If the temperature of the hot water tank  $T_{\text{WS}}(t)$  falls below  $T_{\text{WS}}^{\text{heat}}$ , either the micro-CHP plant or the condensing boiler are required to heat the water in the storage tank before any further water may be extracted for heating purposes. Additionally, the hot water storage possesses a maximum energy  $Q_{\text{WS}}^{\text{max}}$  that can be extracted at most per time slot. Based on these constraints, the amount of energy that is extracted, is defined as

$$Q^{\text{heat}}(t) = \begin{cases} \max(0, \min(Q^{\text{NH}}(t), Q_{\text{WS}}^{\text{max}})) & \text{if } T_{\text{WS}}(t) > T_{\text{WS}}^{\text{heat}} \\ 0 & \text{else.} \end{cases} \quad (7.11)$$

A model for the temperature of the warm water storage is presented next. The temperature of the hot water storage  $T_{\text{WS}}(t)$  in time slot  $t$  depends on its temperature  $T_{\text{WS}}(t-1)$  in the previous time slot  $t-1$  and the energy that is fed into and extracted from the storage. These include the heat generated by the micro-CHP plant and the condensing boiler  $Q^{\text{in}}(t) = Q_{\text{CHP}}(t) + Q_{\text{CB}}(t)$ , the extracted heat  $Q^{\text{heat}}(t)$  for heating the building, the energy of the warm water used by the residents  $Q^{\text{warm}}(t)$  and the energy required for maintaining the current temperature of the water storage tank  $Q_{\text{WS}}^{\text{standby}}(t)$ . The sum of these energies is divided

## 7. Real-world Applications

by the product of the water storage volume  $V_{WS}$ , the density and specific heat capacity of water  $\rho^{\text{water}}$  and  $c^{\text{water}}$ <sup>4</sup> to obtain the temperature change:

$$T_{WS}(t) = T_{WS}(t-1) + \frac{Q^{\text{in}}(t) - Q^{\text{heat}}(t) + Q^{\text{warm}}(t) - Q_{WS}^{\text{standby}}(t)}{V_{WS} \cdot \rho^{\text{water}} \cdot c^{\text{water}}}. \quad (7.12)$$

Additionally, The hot water storage possesses a minimum  $T_{WS}^{\text{min}}$  and maximum operation temperature  $T_{WS}^{\text{max}}$ . The storage temperature is required to remain in these bounds. All schedules that result in the violation of these boundaries in at least one time slot are infeasible. Since the hot water storage is not perfectly insulated, its heat dissipates into the building.<sup>5</sup> The energy required to maintain the current temperature of the hot water storage is obtained by

$$Q_{WS}^{\text{standby}}(t) = (T_{WS}(t-1) - T(t-1)) \cdot h_{WS}^{\text{loss}}, \quad (7.13)$$

where  $h_{WS}^{\text{loss}}$  is the heat transfer coefficient of the hot water storage. The complete thermal and scheduling model allows defining the objectives that are optimized.

Residential buildings are provided with electricity and natural gas by a utility company. In the proposed framework, it is assumed that gas can be purchased at a fixed price  $p^g$  and that the price for electricity  $p^b(t)$  varies across time, i.e. in each time slot  $t$  residents may pay a different price per unit for the amount of electricity they buy from the utility. If the amount of electricity generated in the building exceeds the building's consumption, the surplus is sold to the utility. The price per unit of energy sold depends on its source of generation. Electricity is generated in the building by the micro-CHP plant and a PV system. The energy output of both devices at time  $t$  is denoted by  $E_{\text{CHP}}(t)$  and  $E_{\text{PV}}(t)$ . Note that the sign of both quantities is negative to indicate that energy is generated instead of consumed. Electricity generated by a micro-CHP plant and PV system is remunerated at  $p_{\text{CHP}}$  and  $p_{\text{PV}}$ , respectively, per unit. The final price is determined by the share of each generating device in the total amount of electricity produced (Equation (7.14)).

Finally, the model implements the notion of a load limitation mechanism. The utility may be interested in capping peak consumption to guarantee the stability of the electric grid in providing customers with an incentive to limit their energy consumption in times of peak demand. In case the total amount of energy bought exceeds the threshold  $E^{\text{max}}$ , residents are obliged to pay a penalty  $P(t, E(t) - E^{\text{max}})$ . The penalty is also time-variable such as the purchase price of electricity and depends on the excess demand  $E(t) - E^{\text{max}}$ . Let  $J$  be the set of

<sup>4</sup>In this simplified model it is assumed that the specific heat capacity is constant i.e. does not depend on the water temperature.

<sup>5</sup>The loss  $Q_{WS}^{\text{standby}}(t)$  forms part of the heat loss/gain expressed by  $Q^{\text{other}}(t)$ .

## 7.2. Building Energy Management Systems

all devices in the building and  $E(t) = \sum_{j \in J} E_j(t)$  and  $G(t) = \sum_{j \in J} G_j(t)$ . The energy costs at any time slot  $t$  is computed as

$$C(t) = \underbrace{p(t) \cdot E(t)}_{\text{energy costs/renumeration}} + \underbrace{p^g \cdot G(t)}_{\text{gas costs}} + \underbrace{P(t, E(t) - E^{\max}) \cdot 1_{E(t) > E^{\max}}}_{\text{penalty for violation of load limitation}}, \quad (7.14)$$

where

$$p(t) = \begin{cases} p^b(t) & \text{if } E(t) \geq 0 \\ \frac{E_{\text{CHP}}(t) \cdot p_{\text{CHP}} + E_{\text{PV}}(t) \cdot p_{\text{PV}}(t)}{E_{\text{CHP}}(t) + E_{\text{PV}}(t)} & \text{if } E(t) < 0. \end{cases} \quad (7.15)$$

The total energy costs  $C$  are obtained by summing the costs across all time slots:  $C = \sum_{t \in H} C(t)$ , where  $H$  is the set of considered time slots. Negative costs imply that the residents earn money by selling unused electricity.

Carbon dioxide emissions are caused by burning gas in the micro-CHP plant and condensing boiler or by consuming electricity bought from the grid. Each unit of gas causes a fixed emission of  $e^g$ . The electric grid is fed by numerous energy suppliers. Suppliers generate electricity using different technologies that potentially involve the emission of carbon dioxide. The proposed framework assumes that there exists a signal  $e^b(t)$  that states the average amount of carbon dioxide emitted for generating a unit of electricity that can be procured from the grid in time slot  $t$ . The carbon dioxide emitted in  $t$  is the sum of the emission of electricity bought from the grid and of the natural gas burned locally:

$$B(t) = e^b(t) \cdot E(t) \cdot 1_{E(t) > 0} + e^g \cdot G(t). \quad (7.16)$$

Total carbon dioxide emissions are summed across the entire optimization horizon to obtain the overall carbon dioxide emissions  $B = \sum_{t \in H} B(t)$ .

The last objective consists of the average deviation from a target lower  $T^l$  and upper temperature  $T^u$  set by the residents for the building. Whenever the temperature in the building falls below  $T^l$  or rises above  $T^u$  discomfort incurs:

$$D(t) = (T^l - T(t)) \cdot 1_{T(t) < T^l} + (T(t) - T^u) \cdot 1_{T(t) > T^u}. \quad (7.17)$$

The average thermal discomfort is obtained by summing the discomfort of all time slots and dividing the result by the number of time slots of the optimization horizon:  $D = \sum_{t \in H} D(t) / |H|$ . To prevent extreme temperature drops and spikes, residents are allowed to set a minimum  $T^{\min}$  and a maximum temperature  $T^{\max}$  that defines bounds in which the building temperature has to stay in all time slots. Any schedule that results in the building temperature to move beyond these bounds is considered infeasible. An illustration of the thermal discomfort is provided in Figure 7.8. An overview of all relevant parameters,

## 7. Real-world Applications

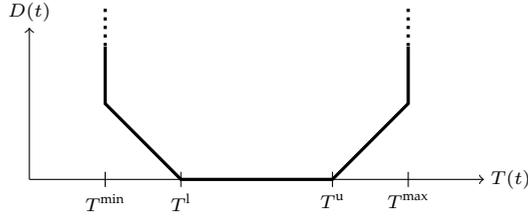


Figure 7.8.: Illustration of how thermal discomfort is computed. If the building temperature  $T(t)$  is within the comfort threshold  $[T^l, T^u]$  no discomfort incurs. If the temperature lies either in  $[T^{\min}, T^l)$  or  $(T^u, T^{\max}]$  the discomfort increases linearly with respect to the distance to the comfort threshold. Any schedule that results in the temperature falling below  $T^{\min}$  or rising above  $T^{\max}$  is considered to be infeasible, which is indicated by a discomfort of positive infinity.

variables and functions is given in Table F.6. The complete problem formulation is given in Equation (7.18a).

$$\min_{s^j, p^j, a^j, n_j} (C, B, D) \quad \text{s.t.} \quad (7.18a)$$

$$T^{\min} \leq T(t) \leq T^{\max} \quad \forall t \in H \quad (7.18b)$$

$$T_{\text{WS}}^{\min} \leq T_{\text{WS}}(t) \leq T_{\text{WS}}^{\max} \quad \forall t \in H \quad (7.18c)$$

$$r_j + \sum_{i=1}^{n_j} s_{j,i} + p_{j,1} \leq d_j \quad \forall j \in J \quad (7.18d)$$

By rescheduling devices within the optimization horizon, different costs, emissions and discomfort may occur. There exists an inherent tradeoff between the three objectives. Minimizing costs favors executing devices during periods in which electricity is cheap. Such periods, however, may not be characterized by low carbon dioxide emissions of energy sold by the utility. Heating the building in winter and cooling it in summer requires the operation of the air-conditioning or the condensing boiler and the micro-CHP plant. All three devices either consume electricity or natural gas, which leads to an increase in costs and emissions.

Table 7.7.: Overview of the problem instances in the BEMS study.

	RW1/RS1	RW2/RS2	RW3/RS3	Temporal flexibility
Washing machine	deferrable	deferrable	deferrable and load-flexible	08:00 - 17:00
Tumble dryer	deferrable	interruptible	interruptible and load-flexible	18:00 - 21:00
Dishwasher	deferrable	interruptible	interruptible and load-flexible	10:00 - 18:00

## 7.2.2. Optimal Operating Schedules in Residential Buildings

The computational study considers an optimization horizon of 24 hours with time slots of one minute length and full information about all relevant parameters within the optimization horizon, i.e. there exist perfect weather predictions for the next 24 hours and there is no uncertainty about the behavior of the residents. Such simplifications may appear unrealistic within a practical scenario, however they are a necessary requirement for allowing a meaningful analysis of the optimization results. Otherwise, schedules that are optimal ex-ante, may turn out to be not optimal ex-post, e.g. if the schedule expected a different residential behavior. Such scenarios should be dealt within the realm of robust optimization. Instead, the scope of this study is the exploration of tradeoffs that occur between objectives as the result of selecting different schedules and how the selection process can be simplified for the residents

Six different scenarios – RW1-RW3; RS1-RS3 – are considered each of which consists of finding optimal schedules for a single residential household (Table 7.7). In each scenario, the residential building possesses a micro-CHP plant, a condensing boiler, a hot water storage tank, a PV system, a dishwasher, a washing machine and a tumble dryer. The six scenarios comprise two different weather settings: a cold winter day in January (RW1-RW3) and a hot summer day in June (RS1-RS3). The weather strongly affects the demand for heating and cooling and the generation of electricity by the PV system, which is why its effect on the optimization outcome is studied. Air-conditioning is only considered in the summer scenario. The scenarios are further categorized by the type of flexibility the dishwasher, washing machine and tumble dryer exhibit: deferrable (RW1/RS1), deferrable/interruptible (RW2/RS2) and deferrable/interruptible and load-flexible (RW3/RS3). The effect of flexibility on the optimization results can therefore be assessed separately.

Load profiles of the washing machine, dishwasher and tumble dryer were obtained from measurements in the laboratory environments of the KIT Energy Smart Home Lab (ESHL)<sup>6</sup> Technical parameters for the HVAC devices were taken from real products (Table 7.8) if possible and missing data was amended

<sup>6</sup><http://www.organic-smarthome.org>. Accessed 04.12.2017.

## 7. Real-world Applications

Table 7.8.: Overview of HVAC devices used in the BEMS study.

Appliance	Product	Manufacturer
Micro-CHP Plant	<i>ecoPower 1.0</i>	<i>Vaillant</i>
Condensing boiler	<i>Logamax plus GBH172</i>	<i>Buderus</i>
Hot water storage tank	<i>VITOCCELL 100-E</i>	<i>Viessmann</i>

by proper considerations. Load profiles are documented in Appendix F and in the digital appendix.

A four person household served as blueprint for modeling the residential building. The floor space and ceiling height were chosen as  $A = 130 \text{ m}^2$  and  $h = 2 \text{ m}$ . The German standard load profiles of households  $H0$  provided by the *German Association of Energy and Water Industries* was chosen to model the electricity consumption of non-optimizable devices. Time-variable electricity prices correspond to those employed in the project *iZeus* [Dal13]. The same load limitation signal as employed by Allerdig et al. [All14] was used. The penalty for exceedance was set as paying twice the current price. A natural gas price of 9.16 Cent per kWh was chosen, which reflect current tariffs in Germany. Feed-in tariffs are based on the *German Renewable Energy Act* with 12.56 Cent per kWh for PV systems and 8.53 Cent per kWh for micro-CHP plants. Carbon dioxide emissions of electricity obtained from the grid is based on data from the *Fraunhofer Institute of Solar Energy*<sup>7</sup>. The hot water consumption profile was obtained from *Directive 2010/30/EU* of the European Commission<sup>8</sup>. Lower and upper bounds for the temperature were set to  $T^l = 19.5 \text{ }^\circ\text{C}$  and  $T^u = 20.5 \text{ }^\circ\text{C}$  in January and to  $T^l = 21.5 \text{ }^\circ\text{C}$  and  $T^u = 22.5 \text{ }^\circ\text{C}$  in June. The minimum and maximum temperatures inside the building were set to  $T^{\min} = 18 \text{ }^\circ\text{C}$  and  $T^{\max} = 25 \text{ }^\circ\text{C}$  in all scenarios. Outside temperatures were obtained from the online weather portal *wetter.com* at an hourly resolution in Karlsruhe for the dates 7.1.2015 (RW1-RW3) and 4.6.2015 (RS1-RS3). Solar irradiance was measured in the *ESHL*. The code of the simulation is available online [Bra].

As stated in the introduction, the main focus of this study is identifying schedules that are local scalarization optima. For this purpose, the algorithmic framework that was introduced in Algorithm 11 is applied. Pareto front approximations in step 1 of Algorithm 11 may be obtained by various methods – mathematical programming, metaheuristics or a combination of the two. In optimization, it is usually advisable to first try an exact approach to solve an optimization problem before applying a heuristic, since the latter are not guaranteed to find an optimum within a certain error bound. If no canonical form

<sup>7</sup>[https://www.energy-charts.de/power\\_de.htm](https://www.energy-charts.de/power_de.htm). Accessed 04.12.2017.

<sup>8</sup><http://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex:32013R0812>. Accessed 04.12.2017.

can be formulated that is an adequate representation of the physical reality or if the mathematical program cannot be solved within a reasonable time frame, heuristics should be applied to solve the problem.

Scheduling problems such as the one presented in Section 7.2.1 are usually expressed as integer programs (IPs).<sup>9</sup> In an IP, decision variables are integers and objective functions and constraints are linear.<sup>10</sup> IPs are known to be NP-hard [Kar72] implying that there is no known algorithm that solves a general IP within polynomial runtime. As the number of decision variables increases, IPs usually become increasingly difficult to solve within a reasonable time frame and mathematical programs often do not find close approximations of optima at all. In such cases, metaheuristics often outperform mathematical programming techniques [APSS12]. Additionally, there exist few mathematical programming techniques for three objective IPs (see [GEF16, Sec. 19.4.2]) On top of that, these techniques do not perform well for a larger number of decision variables [PGE09]).

For these reasons, an evolutionary approach was chosen to approximate the Pareto fronts of the six scenarios. Integer variables were encoded as real numbers, i.e. the values stored in the genome were rounded to obtain the devices' schedule. In general, integers can either be directly used as values in the genome or encoded as reals or binary strings. These three options possess different advantages and disadvantages. Indirect encodings such as the real and binary string representation possess the advantage that there exist efficient recombination and mutation operators for both, whereas search operators that operate directly on integers are not well explored [ES<sup>+</sup>03].

On the other hand, the performance of search operators operating on indirect encodings depend on the composition of the problem domain. Binary strings possess a fixed length of  $a$  bits, which allows the representation of  $2^a$  different integers. If there are  $b$  different integers to represent, then  $a$  must be chosen such that  $2^a \geq b$ . This implies that there exist  $2^a - b$  strings that do not map to a feasible value of the problem domain. Such strings must be either repaired or discarded if they are the result of a recombination or mutation operation.

Search operators on real-valued integer representations might be inefficient if the number of feasible integers is small, since rounding may result in small changes in the genome not translating to the decoded integer variable. For example, if there exist only two integers  $\{0, 1\}$  any mutation operation needs to make the encoded variable either smaller or greater than 0.5 to change the value

---

<sup>9</sup>The proposed problem as presented in Section 7.2.1, however, is not presented in the canonical form of an IP.

<sup>10</sup>There exists variants of IPs that, for example, also allow real-valued decision variables – mixed integer linear programs (MILPs) – or non-linear constraints and objective functions – non-linear integer programs – among others.

## 7. Real-world Applications

Table 7.9.: BEMS study – IGD. Median and IQR (as subscript) results after 50 000 function evaluations. Best performances are colored in green, second best in blue. NaN implies that the algorithm did not find any feasible candidate solution.

	ESPEA	IBEA	MOEAD	NSGA-II	NSGA-III	SMPSO	SMS-EMOA	SPEA2
RW1	4.86 <sub>2.17</sub>	0.16 <sub>0.39</sub>	4.66NaN	1.47 <sub>1.94</sub>	1.67 <sub>2.42</sub>	0.07 <sub>1.21</sub>	2.92 <sub>2.47</sub>	1.55 <sub>2.01</sub>
RW2	11.94 <sub>3.83</sub>	0.23 <sub>1.39</sub>	10.36NaN	3.58 <sub>4.83</sub>	5.58 <sub>6.19</sub>	0.12 <sub>0.36</sub>	7.12 <sub>5.10</sub>	4.27 <sub>5.92</sub>
RW3	5.00 <sub>1.99</sub>	0.22 <sub>0.55</sub>	1.71NaN	1.81 <sub>2.18</sub>	1.54 <sub>2.49</sub>	0.05 <sub>1.73</sub>	3.23 <sub>2.16</sub>	1.63 <sub>2.58</sub>
RS1	0.07 <sub>0.02</sub>	0.09 <sub>0.01</sub>	NaNNaN	0.08 <sub>0.02</sub>	0.08 <sub>0.01</sub>	0.06 <sub>0.02</sub>	0.08 <sub>0.01</sub>	0.08 <sub>0.02</sub>
RS2	0.07 <sub>0.02</sub>	0.08 <sub>0.01</sub>	NaNNaN	0.08 <sub>0.02</sub>	0.08 <sub>0.01</sub>	0.06 <sub>0.01</sub>	0.08 <sub>0.02</sub>	0.08 <sub>0.01</sub>
RS3	0.09 <sub>0.03</sub>	0.10 <sub>0.01</sub>	NaNNaN	0.11 <sub>0.02</sub>	0.11 <sub>0.01</sub>	0.07 <sub>0.02</sub>	0.11 <sub>0.02</sub>	0.10 <sub>0.03</sub>

of the decoded integer. Instead, bit-flip mutation would ensure that a mutation always translates to a change in the decoded decision variable. In the proposed problem, an optimization horizon of 24 hours at a one minute resolution results in 1440 time slots, which is a sufficiently large number for SBX and polynomial mutation to perform well.

Using a real-encoding for the decision variables allows reusing the same algorithms that have been studied in Sections 5.3.2 and 7.1.2 for approximating the Pareto fronts: ESPEA (Algorithm 1), IBEA (Algorithm 6), MOEAD (Algorithm 9), NSGA-II (Algorithm 4), NSGA-III (Algorithm 10), SMPSO (Algorithm 7), SMS-EMOA (Algorithm 8) and SPEA2 (Algorithm 5). The algorithms were configured in the same manner as in the previous two studies (Table E.2). A population size of 100 and 50 000 function evaluations were chosen, since it was observed that the populations of the algorithms became evolutionary stable after 50 000 function evaluations had elapsed.

IGD is used as indicator to evaluate the algorithms' performances. Reference Pareto fronts for computing IGDs were generated for each problem instance from the Pareto front approximations of every algorithm across all 100 runs. Table 7.9 shows the median IGD after 50 000 function evaluations for each algorithm and scenario. There exists a notable performance difference between the summer and winter scenarios. The results for the IGD suggest that the majority of the algorithms appear to not perform very well in the winter scenarios. In the median run, the approximations found by ESPEA, MOEAD, NSGA-II, NSGA-III, SMS-EMOA and SPEA2 appear to be far away from the true Pareto front as indicated by the large IGD. Only IBEA and SMPSO manage to obtain satisfactory results. In the summer scenarios, all algorithms exhibit a similar performance with the exception of MOEAD, which is not able to find a single feasible solution within the median number of runs. SMPSO performs best on all problem instances. IBEA delivers the second best results in the winter scenarios and ESPEA obtains the second best performance in the summer scenarios.

Table 7.10.: Extreme objective values obtained among Pareto optimal points in the BEMS study.

	Cost/ct			Emissions/g			Discomfort/°C		
	min	max	diff	min	max	diff	min	max	diff
RW1	1936	1964	27	49689	49832	144	0	0.02	0.02
RW2	2009	2051	42	49733	49801	68	0	0.02	0.02
RW3	2008	2056	121	49707	49828	121	0	0.02	0.02
RS1	123	1951	1828	9538	49829	40291	0	0.72	0.72
RS2	112	2030	1918	9415	49799	40384	0	0.73	0.73
RS3	126	2029	1903	9735	49801	40065	0	0.61	0.61

A possible explanation for the large performance difference between summer and winter scenarios can be found in the magnitude of the extreme values of the Pareto fronts. Recall that IGD normalizes objective values based on the ideal and nadir point.<sup>11</sup> Table 7.10 reveals that the difference between maximum and minimum objective values is much larger in the summer compared to the winter scenarios. This can be explained by a higher solar irradiance in the summer facilitating schedules, in which the majority of the electricity consumed in the building is supplied by the PV system. The small differences between maximum and minimum objective values in winter leads to normalization overstating the distance of any objective vector to the Pareto front in comparison to the summer scenario.

Further reasons for the observed performance differences between the algorithms may be found by analyzing the Pareto fronts of the problem instances. Figures 7.9 and 7.10 depict the Pareto fronts of the six problem instances. The shapes of the Pareto fronts are more similar within the same season and differ largely between seasons. The flexibility of the household appliances appears to have little effect on the optimization outcome. Most notably, the Pareto fronts of the summer scenarios are degenerate as the objectives costs and emissions are aligned. These fronts, however, feature a stark tradeoff between costs/emissions and thermal discomfort. In winter, there exist tradeoffs between all three objectives instead. Summer and winter scenarios have in common that their Pareto fronts are disconnected. In winter, the fronts mainly consist of three vertically disconnected curves and in summer vertically disconnected lines form the Pareto front. This may explain, why SMP SO is the top performing algorithm. As observed in Section 5.3.2, SMP SO exhibits a notable performance drop between two and three objective problems compared to other algorithms. Since the Pareto fronts of the proposed scenarios mainly consist of disconnected lines and curves, these problems are more similar to two ob-

<sup>11</sup>Normalization is mandatory to give each objective equal weight in the distance computation in the IGD metric.

## 7. Real-world Applications

jective than three objective problems for niching mechanisms distributing solution uniformly across the Pareto front. This means that niching mechanisms that perform well for two objectives might possess an advantage on these six problems over three objective problems that possess a non-degenerate, continuous front.

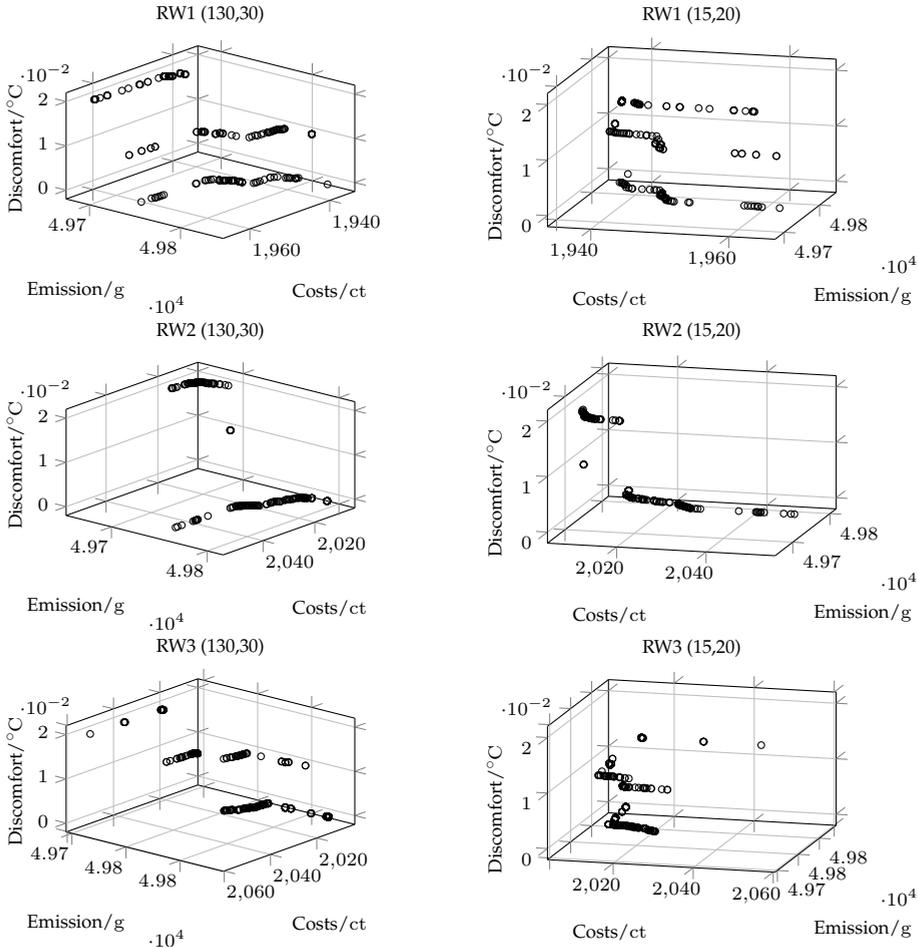


Figure 7.9.: Pareto fronts of the problems RW1, RW2 and RW3. The values in parenthesis indicate azimuth and elevation of the perspective.

A statistical analysis is performed to test the performance differences for significances (see Table G.18 for the results of the Anderson-Darling and Kruskal-Wallis tests). The results in Table 7.11 show that ESPEA is outperformed with

## 7.2. Building Energy Management Systems

Table 7.11.: BEMS study – IGD. The table shows  $p$ -values of the post-hoc analysis. Green cell color indicates that ESPEA outperforms the corresponding algorithm with confidence at a 95 % level, the blue color without confidence. Red cell color indicates that ESPEA is outperformed by the corresponding algorithm with confidence at a 95 % level.

	IBEA	MOEAD	NSGA-II	NSGA-III	SMPSO	SMS-EMOA	SPEA2
RW1	0.0000	0.0004	0.0000	0.0000	0.0000	0.0034	0.0000
RW2	0.0000	0.0001	0.0000	0.0000	0.0000	0.0004	0.0000
RW3	0.0000	0.0000	0.0000	0.0000	0.0000	0.0002	0.0000
RS1	0.0000	0.0000	1.0000	0.0607	0.0000	0.0037	1.0000
RS2	0.0000	0.0000	0.7955	0.1489	0.0000	0.3934	0.9989
RS3	0.9877	0.0000	0.0033	0.0000	0.0000	0.0013	0.9960

confidence by all other algorithms in the winter scenarios. In the summer scenarios, ESPEA has already been established as the second-best performing algorithm. However, in only about half of the comparisons the performance difference is significant. This observation reflects the fact that the IGD achieved by the algorithms is very similar across the summer scenarios.

ESPEA’s rather mediocre performance may be explained by another observation. Preliminary studies have revealed that the majority of the candidate solutions in the initial population represent infeasible schedules. This is likely to be caused by the latin hypercube sampling, which tries to distribute the initial population uniformly across the search space. Although ESPEA is able to handle constrained optimization problems, the algorithm was not specifically designed to execute an efficient search strategy if large portions of the search space are infeasible or even the entire initial population is infeasible. If the entire initial population is infeasible, the individual that exhibits the least constraint violation becomes the sole archive member.<sup>12</sup> The archive can only grow, when feasible schedules are found. This is likely to impede the search in early iterations, since only a single archive member can serve as parent to generate new candidate solutions. ESPEA’s performance on constrained optimization problems could be improved by applying more sophisticated constraint-handling techniques [Coe02] or by redesigning the algorithm to also archive infeasible candidate solutions (e.g. [SS13]). Such considerations, however, go beyond the scope of this work.

In a live system, residents would set a target temperature  $T^{\text{set}}$ , a comfort threshold  $[T^{\text{l}}, T^{\text{u}}]$  and the temporal degrees of freedom – release time  $r_j$  and deadline  $d_j$  – of each optimizable appliance that is not an HVAC device. Using predictions for the weather, non-optimizable load and warm water consumption, the

<sup>12</sup>The sum of absolute constrained violations was applied as selection criterion in all algorithms to compare infeasible solutions to each other.

## 7. Real-world Applications

BEMS would perform a multi-objective optimization and present the residents with the Pareto front approximation. The resident would then be required to act as DM and select a preferred schedule among the Pareto optimal set that is implemented.

Choosing a preferred schedule from a visual or tabular representation of the Pareto front approximation is a non-trivial task [GEF16]. The Pareto fronts of the six scenarios exhibit multiple convex and concave bulges and mainly consist of disconnected lines and curves. These irregular shapes make it difficult to identify a single schedule that is particularly attractive to implement. The notion of local preference optima, however, can be applied to reduce the large Pareto optimal set to a small number of candidate schedules. The reduced candidate set contains only the most interesting alternatives, since local preference optima represent the best choice in their immediate neighborhood in the objective space.

In the live system, the residents would choose a scalarization function that is a suitable representation of their preferences. Instead of showing the Pareto front approximation to the residents, the BEMS would perform Step 2 and 3 of Algorithm 11 to approximate the local scalarization optima of the Pareto front. The residents would then be presented with the local scalarization optima and select a preferred schedule. The presentation can take place visually by plotting the Pareto front and the scalarization optima in a single graph or by listing the optima in a tabular environment.

To illustrate this process, local scalarization optima are obtained from the set of the combined nondominated solutions across all algorithms and runs. TS (Algorithm 12) is executed to determine the basins of attraction (Definition 60). The candidate solution in each cluster that possesses the smallest scalarization value is chosen as estimate of a local scalarization optimum. The local optimization step is skipped to avoid local search escaping its assigned basin.

Since the Pareto front approximations obtained by combining the nondominated solutions across all runs and algorithms are not uniform, clustering algorithms such as TS may overestimate the number of basins. This may lead to the identification of putative scalarization optima, which are essentially FPs (Definition 68). In order to avoid an accumulation of such FPs, an additional filtering mechanism is applied to crop the number of identified scalarization optima. Whenever the distance of two putative scalarization optima is smaller than 0.1 in the normalized objective space, the optimum that possesses the smaller scalarization value of the two is retained and the other one is eliminated.

Table 7.12 lists the number of scalarization optima obtained for different scalarization functions and neighborhood sizes  $k^c$ . Although  $k^c = 6$  has been identi-

## 7.2. Building Energy Management Systems

Table 7.12.: Number of local scalarization optima obtained after applying TS to identify basins of attraction for different scalarization functions. Scalarization optima are selected as those elements that possess the smallest scalarization value in each cluster generated by TS. Results are shown for choosing different neighborhood sizes  $k^c$  and compared to the size of the Pareto front approximation from which clusters are computed.

$k^c$	Front	Sum			Nash			Angle			Tradeoff		
		6	10	20	6	10	20	6	10	20	6	10	20
RS1	176	6	5	2	5	4	2	5	4	2	6	5	3
RS2	188	4	4	4	4	3	3	3	3	3	3	4	4
RS3	119	5	5	4	4	4	3	4	4	3	5	5	4
RW1	328	13	11	10	13	11	10	11	8	7	14	11	9
RW2	260	7	7	7	6	5	6	7	5	5	8	5	5
RW3	449	13	13	13	11	11	7	10	8	7	12	11	8

fied in Section 6.3.1 as optimal parameter choice for finding all basins of attraction, increasing  $k^c$  may prove beneficial if the residents want to further reduce the set of candidate schedules. By increasing the neighborhood size, smaller clusters are usurped into bigger clusters. Thereby, the total number of clusters identified by TS decreases, which leads to the potential elimination of scalarization optima. Reducing the number of candidate schedules may facilitate a simplification of the selection process if the Pareto front possesses a large number of scalarization optima. The results in Table 7.12 show that the number of optima found decreases as  $k^c$  increases.

As Table 7.12 reveals, restricting the Pareto front approximation to local scalarization optima greatly reduces the number of candidate schedules. The DM is thereby enabled to make her selection only among those schedules that are most relevant to her. To illustrate the benefit of this approach, Figures 7.11 to 7.14 show the Pareto fronts and the scalarization optima of the sum of objectives, the Nash bargaining solution, angle utility and tradeoff utility, respectively. Depicting the scalarization optima in conjunction with the Pareto front allows fathoming the tradeoffs that occur between the different schedules. As explained before, the Pareto fronts of the six problems are mostly segmented into lines and curves. The scalarization optima show those schedules that are most desirable in these segments according to the chosen scalarization function.

The Pareto fronts of the winter segments can be divided into three layers each of which is characterized by a distinct thermal discomfort level of 0.0, 0.01 and 0.02. The four scalarization functions put a different emphasis on the three segments across the three scenarios. In RW1, the sum of objectives and the Nash bargaining solution identify more optima on the layer with the highest

## 7. *Real-world Applications*

thermal discomfort. As the flexibility of the household appliances in the winter scenario increases, more optima are identified featuring no thermal discomfort at all. Increasing the flexibility therefore seems to increase the number of low thermal discomfort optima.

The optima identified in the summer scenarios occur at similar positions across the different scalarization functions. These positions and the corresponding schedules can be divided into three categories: 1) no thermal discomfort with high costs and emissions; 2) minor thermal discomfort with small costs and emissions; 3) small thermal discomfort with minor costs and emissions. The DM may possess different preferences for the three objectives, making schedules of all three categories eligible candidates for selection. The results of the summer scenarios illustrate the advantages of the local preference optima approach for solving MOOP. Local scalarization optima narrow the search results to a limited number of schedules that are most interesting to the DM. As indicated by the three categories, these schedules are substantially different from each other with respect to their objective values, such that the DM is presented with true alternatives that result in noticeable differences if implemented in the live system. Giving the DM multiple, substantially different options to choose from, enables her to easily understand the tradeoffs that occur between the objectives without being required to analyze the composition of the entire Pareto front. By presenting the DM with multiple options she is enabled to make her own informed choice that may differ from the solution obtained by a global optimization algorithm, because knowledge of the local optima may change her preconceived preferences.

It would be desirable to quantify how well the methodology proposed in Chapter 6 is able to find the local scalarization optima of the six scenarios. Conducting a quantitative analysis as done in Section 6.3, however, is difficult, since the true Pareto fronts of the problem instances are unknown. The nondominated points obtained by combining the search results of every algorithm across all runs are only an estimate of the true Pareto front. Since the algorithms that are considered in the study are heuristics and no closed form description of the Pareto fronts exist, no formal statement can be made about how well these points represent the true front. As explained in Section 6.3.1 computing local scalarization optima and the corresponding basins of attraction of three objective problems is already challenging even if a closed-form description of the Pareto front is available. Computing optima and basins of a Pareto front, whose exact composition is unknown, is unarguably even harder if not impossible.

The issue outlined in the previous paragraph is a general problem of benchmarking heuristics in the context of SOO and MOO if the global optima or the Pareto front is unknown. The IGD indicator using a reference front consisting of all nondominated points across all algorithms and runs is still a viable means

## 7.2. Building Energy Management Systems

Table 7.13.: BEMS study – IGD. Median and IQR (as subscript) results after 50 000 (Baseline) and 900 000 (Exhaustive) function evaluations of SMP SO. Best performances are colored in green.

	Exhaustive	Baseline
RW1	1.25e-1 <sub>1.41e-1</sub>	2.42e-1 <sub>1.19e1</sub>
RW2	8.25e-2 <sub>1.05e-1</sub>	2.75e-1 <sub>4.28</sub>
RW3	1.24e-1 <sub>1.13e-1</sub>	3.18e-1 <sub>8.63</sub>
RS1	6.82e-2 <sub>6.75e-2</sub>	1.72e-1 <sub>2.09e-1</sub>
RS2	7.82e-2 <sub>4.24e-2</sub>	4.29e-1 <sub>4.05e-1</sub>
RS3	1.27e-2 <sub>1.79e-2</sub>	1.99e-1 <sub>1.33e-1</sub>

for comparing MOEA performances even if the generated approximations are further away from the front. In SOO, objective values can be compared irrespective of whether the globally optimal value is known. Such an approach is not possible for local scalarization optima. An approximation  $\mathbf{u}$  that features a smaller scalarization value than a given reference optimum  $\mathbf{u}^L$  may either be an improvement over  $\mathbf{u}^L$  or an element of a different basin of attraction that features smaller scalarization values. Using the distance of  $\mathbf{u}$  to  $\mathbf{u}^L$ , on the other hand, is not a suitable indicator, since  $\mathbf{u}$  might indeed be an improvement over  $\mathbf{u}^L$ .

A sufficiently close and uniform approximation of the Pareto front may usually be obtained by performing an exhaustive search, i.e. running an MOEA for a very high number of function evaluations and a large population size. Such an approach was taken to confirm the quality of the Pareto front approximation obtained as the nondominated points computed across all algorithms and runs. As top-performing algorithm, SMP SO was run ten times on all problem instances for 900 000 function evaluations using a swarm size of 300. The exhaustive search manages to improve the existing Pareto front approximation as indicated by the results Table 7.13. However even after 900 000 function evaluations are executed, SMP SO sometimes gets stuck during the search and does not manage to converge to the Pareto front (Figure 7.15). It was therefore concluded that no suitable reference data can be generated for a quantitative study within the chosen experimental environment.

It is imperative to note that benchmarking any component of the framework for approximating local scalarization optima on a real-world application is beyond the scope of this section. Instead, the usefulness of the local scalarization optima notion in decision-making is illustrated. The qualitative analysis conducted in this section is intended to serve this purpose.

7. Real-world Applications

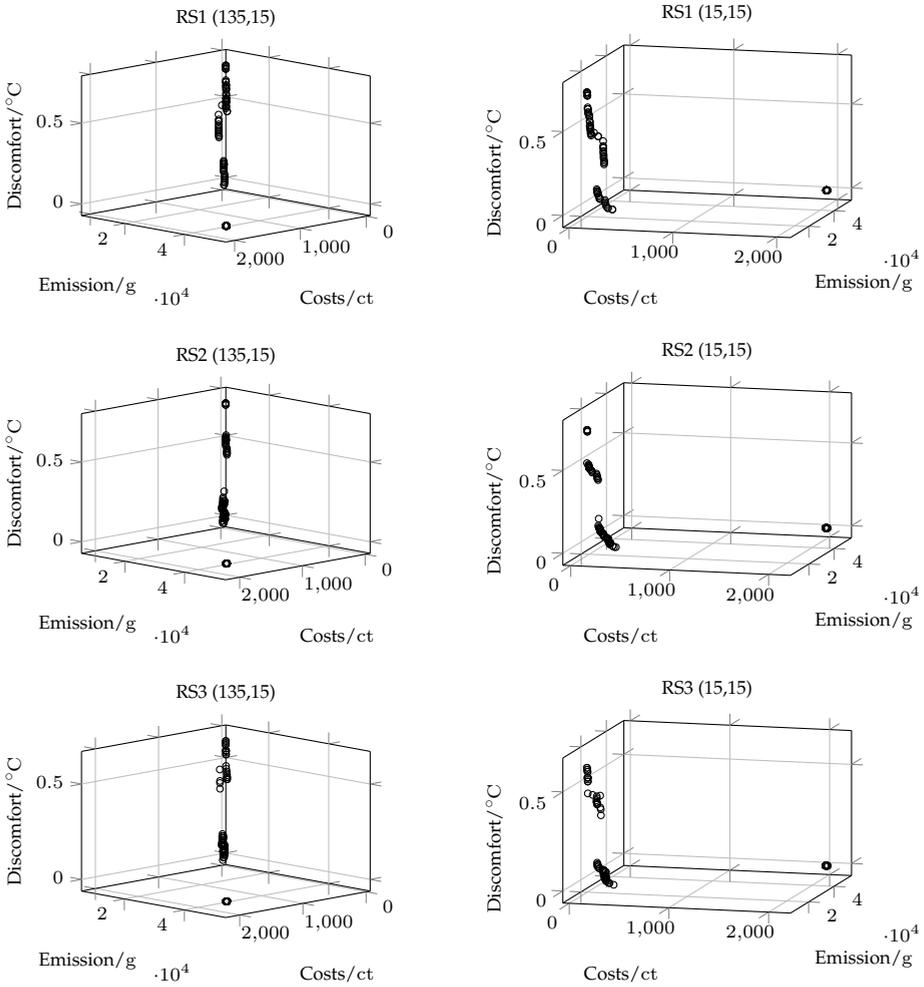


Figure 7.10.: Pareto fronts of the problems RS1, RS2 and RS3. The values in parenthesis indicate azimuth and elevation of the perspective.

## 7.2. Building Energy Management Systems

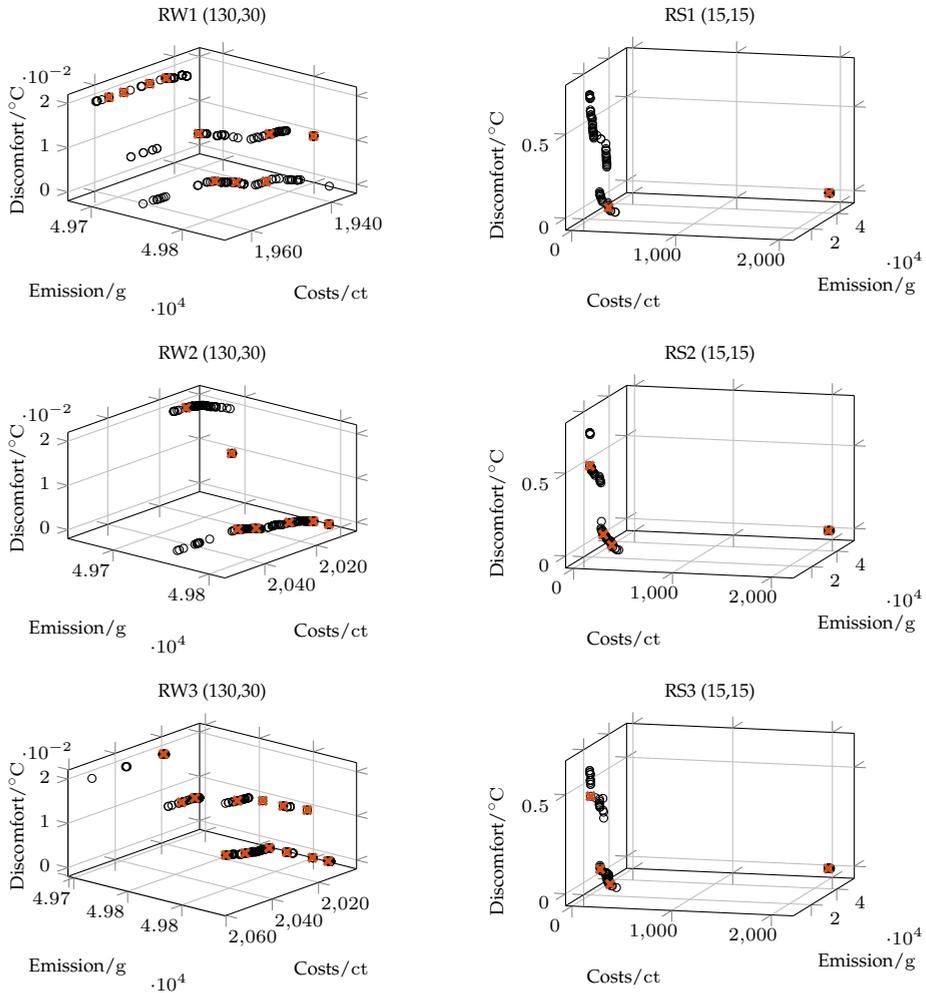


Figure 7.11.: BEMS study – illustration of sum of objectives scalarization minima. Scalarization optima were obtained from a TS clustering with a neighborhood size of  $k^c = 20$ . The values in parenthesis indicate azimuth and elevation of the perspective.

## 7. Real-world Applications

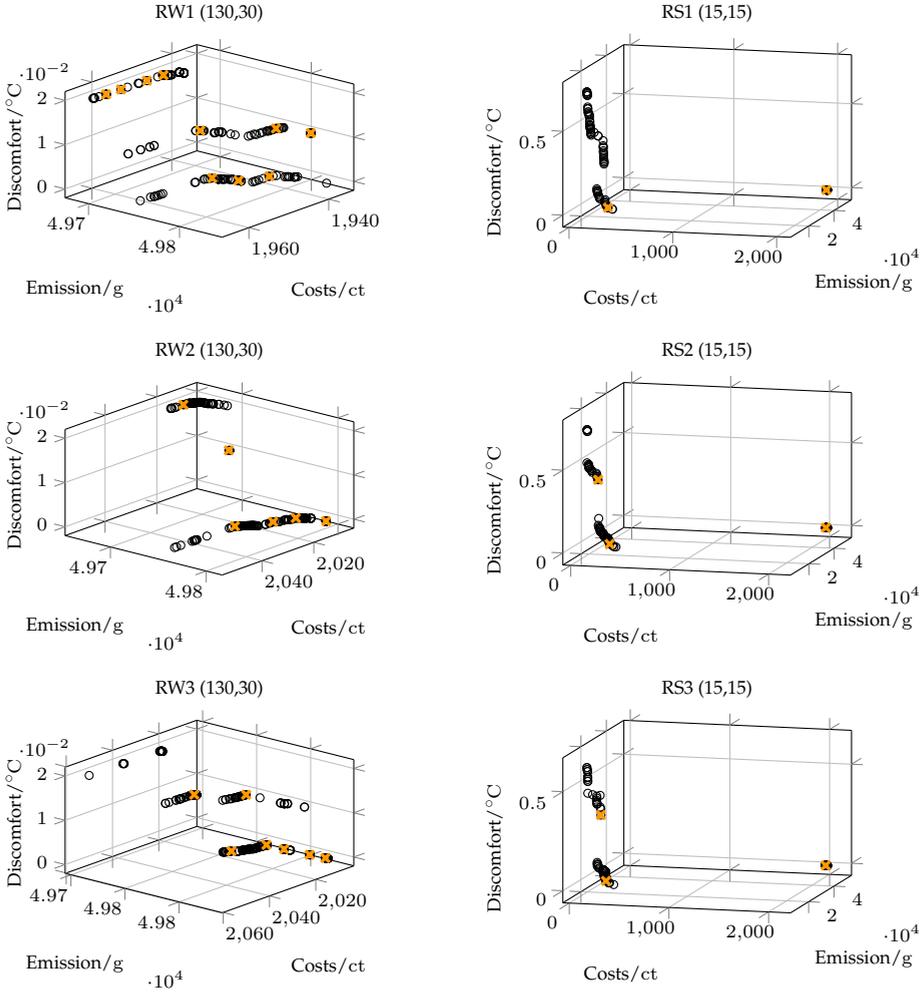


Figure 7.12.: BEMS study – illustration of Nash scalarization minima. Scalarization optima were obtained from a TS clustering with a neighborhood size of  $k^c = 20$ . The values in parenthesis indicate azimuth and elevation of the perspective.

## 7.2. Building Energy Management Systems

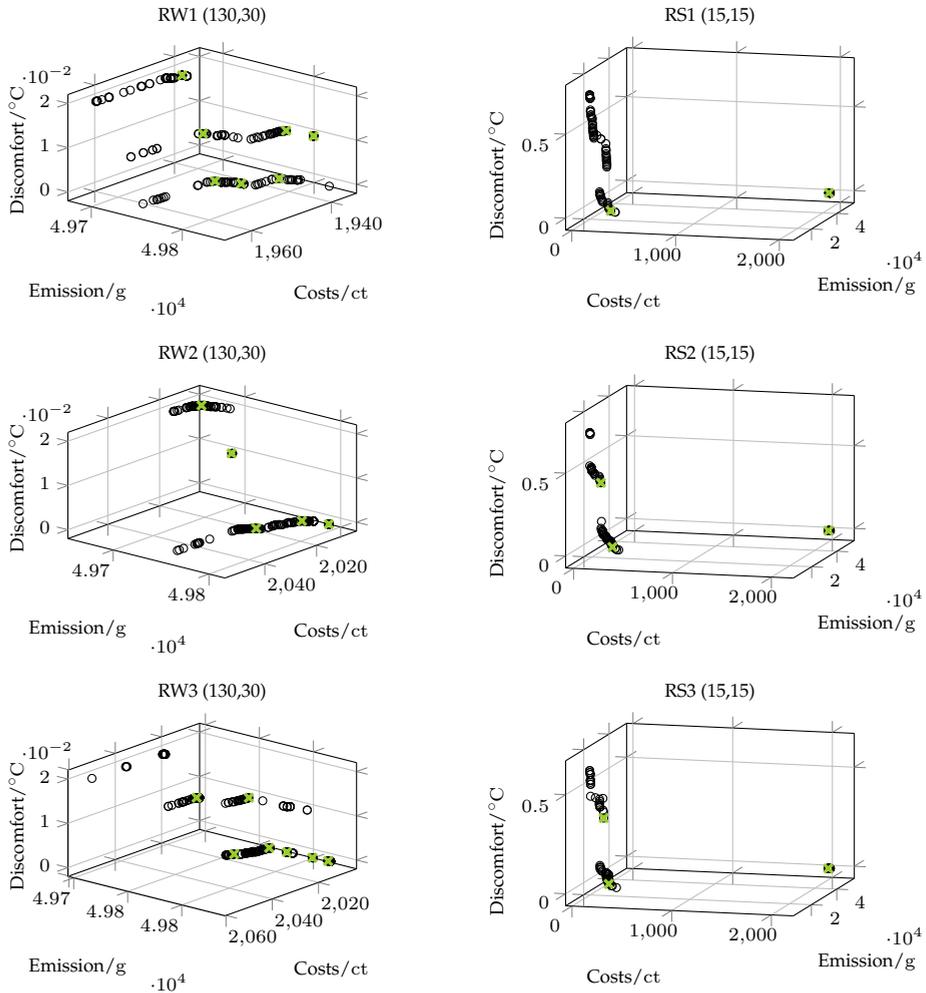


Figure 7.13: BEMS study – illustration of angle utility scalarization minima. Scalarization optima were obtained from a TS clustering with a neighborhood size of  $k^c = 20$ . The values in parenthesis indicate azimuth and elevation of the perspective.

## 7. Real-world Applications

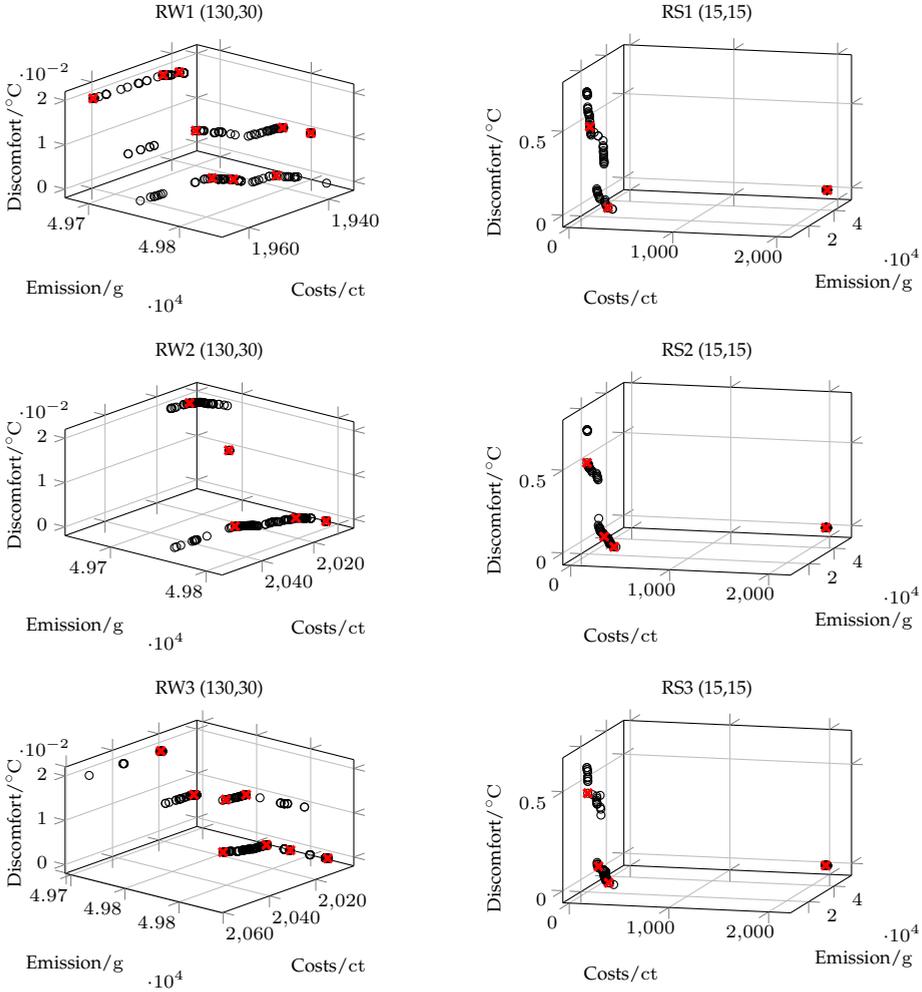


Figure 7.14.: BEMS study – illustration of tradeoff utility scalarization minima. Scalarization optima were obtained from a TS clustering with a neighborhood size of  $k^c = 20$ . The values in parenthesis indicate azimuth and elevation of the perspective.

## 7.2. Building Energy Management Systems

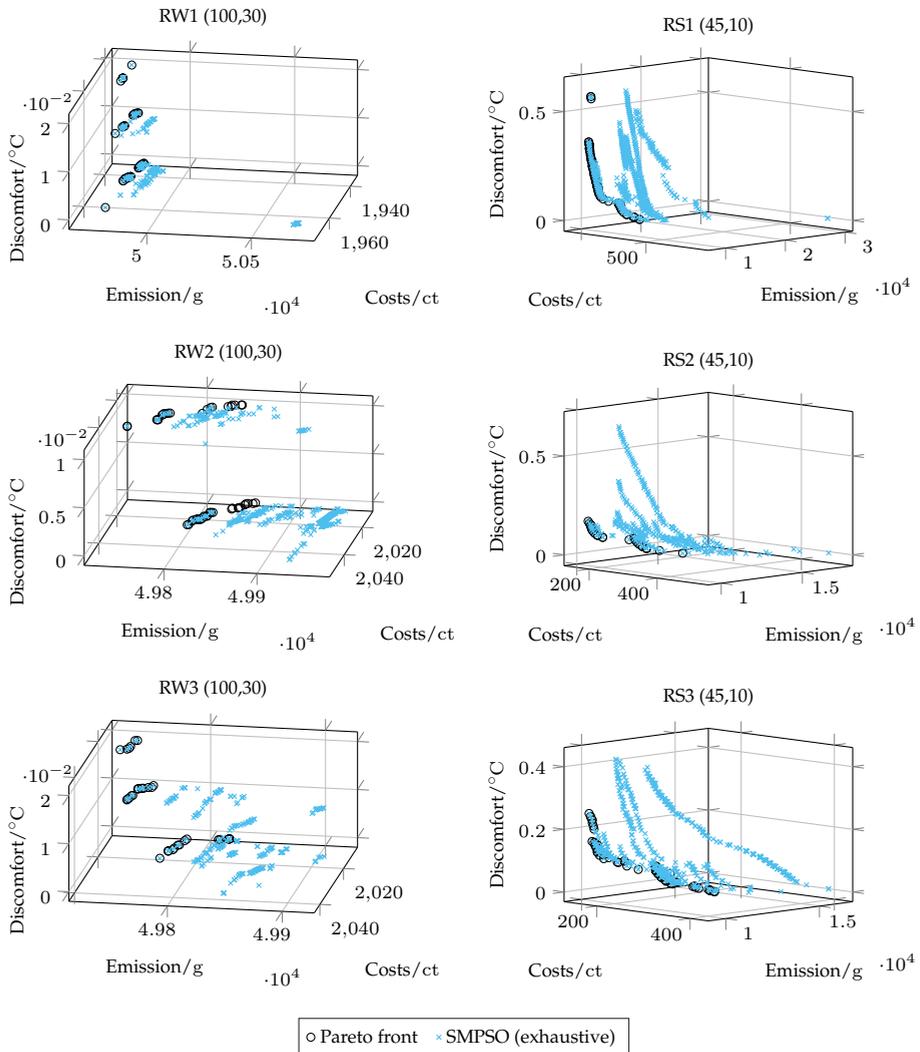


Figure 7.15.: Illustration of the exhaustive search using SMPSO with 900 000 function evaluations. The Pareto front consists of all nondominated candidate solutions obtained across all algorithms and runs.



## 8. Conclusion and Outlook

The study of scalarized preferences and their importance in MOO have been the subject of this thesis. A theoretical and algorithmic analysis of preference has been given and the meaning of scalarization has been explored within this context. The two main contributions of this work consist of a formal model to characterize scalarization functions and the development of two methods to find more than one suitable Pareto optimal solution for implementation from scalarized preference information. This chapter provides a brief conclusion, in which the contributions of this work are discussed. Subsequently, the work is concluded by an outlook on future research topics that emanate from this work.

### 8.1. Conclusion

Six key contributions of this thesis have been presented in the introduction in Section 1.3. In the following, a brief summary is given of how this work has provided these contributions.

A theoretical framework for describing preferences as problem transformations has been developed in Chapter 2. Preference predicates define a preferred subset of the set of Pareto optimal solutions. Within the framework, preferences can be expressed by constraining the domain of the objective function, mapping the codomain of the objective function to another set or amending the order imposed on the codomain of the objective function. Scalarization represents a transformation of the codomain by mapping the vector of objectives to a single real value. It has been shown that, in theory, a preference can be expressed by only applying either one of the three transformation types. A practical example has been given in the form of tradeoff optimality. Tradeoff optimal solution can be obtained by introducing additional constraints to the optimization problem, enforcing a domination relation comparing maximum and minimum objective differences or by identifying the nondominated points of the feasible set computed by transforming the objective space using a decomposition of polyhedral cones.

## 8. Conclusion and Outlook

A system of axioms that represent desirable properties a scalarization function may exhibit has been presented in Chapter 3. These axioms are Pareto compliance – if an objective vector  $\mathbf{u}$  Pareto-dominates another objective vector  $\mathbf{v}$ , then the scalarization value of  $\mathbf{u}$  must be smaller than the one of  $\mathbf{v}$ ; binary independence – scalarization values do not depend on the composition of the feasible objective space; non-extremeness – the scalarization function does not identify an extreme point as global scalarization optimum; contraction consistency – the location of the scalarization optimum does not change if the Pareto front becomes smaller; monotonicity – if the composition of the Pareto front changes such that the ideal value of a single objective  $i$  becomes smaller and all other ideal values remain the same, the value of objective  $i$  at the scalarization optimum must decrease; equity – transfers from a larger to a smaller objective must make the scalarization value smaller; and invariance to scaling – the location of the scalarization optimum is not affected by affine transformations of the objective functions. It has been shown that many scalarization functions fulfill several of the proposed axioms, however often only if specific conditions are met.

An approach to characterizing preferences from an algorithmic perspective by classifying the search result an algorithm delivers has been introduced in Chapter 4. The Pareto front can be described by a mathematical formula as a closed-form description or approximated uniformly by a finite set of points. Preference-driven approximations may approximate preferred subsets of the Pareto front by a finite set of points or identify a single global preference optimum. Two new paradigms have been identified: 1) preference-biased Pareto front approximations approximate the entire front by a finite set of points but put a stronger emphasis on regions that are interesting to the DM; 2) local preference optima represent Pareto optimal solutions that are the most desirable choices in their immediate neighborhood in the objective space. For each of the two paradigms, an algorithm has been developed that utilizes scalarized preference information.

In Chapter 5, a concept has been presented for defining an optimal distribution of points on the Pareto front using scalarized preference information for a fixed number of points. It has been shown that such distributions exist for real-valued and discrete MOOP under very mild conditions. If the scalarization function assigns the same scalarization value to all objective vectors a uniform distribution of points is achieved. A steady-state, archive-based MOEA—ESPEA—has been developed to approximate such preference-biased distributions.

An algorithmic framework for finding local optima of scalarized MOOP has been developed in Chapter 6. The framework consists of the three steps for each of which different methods can be applied: 1) Pareto front approximation,

2) basin identification, 3) local optimization. The Pareto front is approximated to narrow down the search to the subset of the feasible set that contains the scalarization optima. Clustering methods are applied to identify the basins of attraction of the local optima, i.e. the neighborhood around the optimum in which a local search is expected to converge quickly towards the local optimum. The final step consists of executing a local search by using the clusters identified in the previous step as initial population.

Both algorithms have been evaluated on artificial and real-world optimization problems. Different archive-update mechanisms have been tested with ESPEA. It could be shown that eliminating the archive member that contributes least to the distribution of points is most beneficial. ESPEA has been compared to seven other state-of-the-art metaheuristics for obtaining uniform finite set of points approximations of the Pareto front. The results have revealed that ESPEA is the second-best performing algorithm among all eight algorithms and that the performance difference between ESPEA and the top-performing algorithm—SPEA2—is negligible. A qualitative analysis of ESPEA with different scalarization functions has been conducted. The preference-biased Pareto front approximations show that ESPEA is successful in finding more solutions in areas that are interesting to the DM, while locating fewer solutions in those regions that feature high scalarization values. The optimization of the operation efficiency of a CHP plant has been chosen to test ESPEA on a real-world application. The results have shown that ESPEA delivers the best performance among all eight algorithm in generating a uniform approximation of the Pareto front of the problem. ESPEA has also been able to obtain preference-biased approximations of the Pareto front using the operation efficiency as scalarization value. Such an approximation may help the plant operator in choosing an appropriate schedule if she wishes to deviate from the efficiency optimum.

Two different clustering algorithms—TS and NBC—have been tested for basin identification on artificial benchmark problems. The simulation results have shown that their performance greatly depends on the chosen parametrization. Both algorithms exhibit a similar performance using their best configuration. Four local optimization algorithms—CMA-ES, a GA, a HC and a PSA—have been investigated using different parametrizations for finding local scalarization optima. The PSA has been identified as delivering the best results. The performance of all algorithms, however, is strongly influenced by their chosen parametrization. A BEMS optimization problem in a residential building has been presented to assess the usefulness of local scalarization optima from a decision-making perspective. It could be shown, that by focusing the search result on local scalarization optima, the number of solutions can be greatly reduced such that the DM is only presented with the most relevant schedules.

## 8.2. Outlook

The ideas and notions that have been developed within this thesis pave the way to pursuing new research endeavors. Existing preference notions in MOO should be categorized within the framework of describing preferences as problem transformations. This could lead to new insights into how different notions are related to each other. The implications of Theorem 1 should be further explored in a theoretical and practical context. It could be investigated, whether it is possible to design formalisms that allow the transformation of specific types of preferences predicates into each other. For example, it would be interesting to assess whether there exists a generalized approach to develop order predicates for scalarization functions as it has been exemplified for tradeoff optimality in Section 2.3.

The axioms developed in Section 3.2 to characterize desirable properties of scalarization functions are only a modest proposition. Researchers and business practitioners should be asked to evaluate the proposed axioms and describe properties they deem themselves desirable in the context of scalarized preferences. For some of the proposed axioms it has been shown that they apply to specific scalarization functions only under very restrictive circumstances. A general catalog could be developed that summarizes which conditions must be met for a scalarization function to satisfy or violate a given axiom.

The energy concept for defining optimal distributions of points on the Pareto front as presented in Definition 53 tends to overstate the importance of boundary points for three and more objectives. The reason for this behavior is that boundary points possess only neighboring points towards the interior of the front. Modifying the energy concept to remedy the boundary issue would be a promising direction of research, since there exist few well performing niching techniques in many-objective optimization (MaO) that are not reference-point based. Improved constraint-handling techniques could be introduced to ESPEA such that the algorithm is better equipped to handle problems whose search space is largely infeasible.

Multimodal scalarized preferences need to be further explored from a theoretical and algorithmic perspective. More test problems should be designed that are tailored to reflect specific challenges that occur in finding local scalarization optima. It needs to be further investigated how the basin notion as presented in Definition 60 influences the convergence of the local search. Furthermore, analytical tools should be developed to compute basins of problems featuring three and more objectives. Especially, conflicts occurring because of the existence of multiple paths of descent to several local optima need to be addressed in this context. It is also worth investigating, whether the basin notion can be

extended in a meaningful way beyond the Pareto front to the entire feasible objective space. The robustness of the clustering methods to identify basins of attraction with respect to perturbations in the Pareto front approximation needs to be further investigated. As evidenced by the BEMS optimization problem presented in Section 7.2, it is often highly difficult or even intractable to obtain a sufficiently close and uniform approximation of the Pareto front in real-world applications. The performance of the clustering methods with respect to the sample size is also an interesting area of research.

Local search for finding local scalarization optima could also be further investigated. New methodologies that prevent local search to escape its assigned basin should be developed. An interesting direction worth pursuing might be using ideas from tabu search to constrain the feasible set during the local search. The approach presented in Chapter 6 is based on restricting the step size of the search, which might deter the convergence of the search if the initial point or population is still further away from the local optimum. In general, the effect of the quality of the clustering on the local search needs to be further explored.

Local scalarization optima should be specifically examined in the context of MaO. Pareto fronts of four and more objectives are difficult to visualize. The notion of local optima is therefore an ideal approach to select a limited number of Pareto optimal solutions for the DM to choose from. A rigorous computational study needs to be performed to assess the performance of clustering and local search algorithms on problems with more than four objectives. The proposed analysis, however, rests on the development of meaningful test problems and techniques to properly determine basins of attraction and local scalarization optima for benchmarking.



# Appendix



## A. Declaration of Published Work

The content presented in this thesis is partially based on work that has been previously published by this author. Each chapter and section in which previously published work is presented is preceded by a declaration that states its source of origin. The following list contains all publications of this author that have contributed to this thesis. For each publication, the relevant contributions of every author are clarified.

- Marlon Alexander Braun, Pradyumn Kumar Shukla, and Hartmut Schneck. Preference ranking schemes in multi-objective evolutionary algorithms. In Ricardo H.C. Takahashi, Kalyanmoy Deb, Elizabeth F. Wanner, and Salvatore Greco, editors, *Evolutionary Multi-Criterion Optimization*, volume 6576 of *LNCS*, pages 226–240. Springer, 2011.

Shukla presented the idea of modifying existing MOEAs to obtain solutions satisfying M-proper Pareto optimality – a concept first published in [Shu07]. This author developed and implemented the necessary modifications of the algorithms presented in the article, whereas pNSGA-II was only reimplemented based on the description in [SHS10a]. Schneck provided commentary and advice.

- Pradyumn Kumar Shukla, Marlon Alexander Braun, and Hartmut Schneck. Theory and algorithms for finding knees. In Robin C. Purshouse, Peter J. Fleming, Carlos M. Fonseca, Salvatore Greco, and Jane Shaw, editors, *Evolutionary Multi-Criterion Optimization*, volume 7811 of *LNCS*, pages 156–170. Springer, 2013.

This author provided the idea for the notion of proper utility and U-domination. Shukla developed the final mathematical formalization for both concepts. All algorithms were implemented by this author and developed by Shukla. Schneck provided commentary and advice.

- Pradyumn Kumar Shukla and Marlon Alexander Braun. Indicator based search in variable orderings: Theory and algorithms. In Robin C. Purshouse, Peter J. Fleming, Carlos M. Fonseca, Salvatore Greco, and Jane Shaw, editors, *Evolutionary Multi-Criterion Optimization*, volume 7811 of *Lecture Notes in Computer Science*, pages 66–80. Springer, 2013.

## A. Declaration of Published Work

Conceptual work was carried out by Shukla. This author provided the implementation and conducted the experimentation.

- Pradyumn Kumar Shukla, Marlon Alexander Braun, and Hartmut Schmeck. On the interrelationships between knees and aggregate objective functions. In *Proceedings of the 2014 Conference Companion on Genetic and Evolutionary Computation Companion*, GECCO Comp '14, pages 95–96, New York, NY, USA, 2014. ACM.

The publication consists of results developed within the master thesis of this author of which Shukla was the adviser.

- Marlon Alexander Braun, Pradyumn Kumar Shukla, and Hartmut Schmeck. Obtaining optimal pareto front approximations using scalarized preference information. In *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*, GECCO '15, pages 631–638, New York, NY, USA, 2015. ACM.

This author developed ESPEA and the concept of using the notion of electrostatic potential energy to obtain preference-biased Pareto front approximations. All experimentation and analysis was carried out by this author. Shukla and Schmeck provided commentary and advice.

- Marlon Alexander Braun, Thomas Dengiz, Ingo Mauser, and Hartmut Schmeck. Comparison of multi-objective evolutionary optimization in smart building scenarios. In *European Conference on the Applications of Evolutionary Computation*, pages 443–458. Springer, Cham, 2016.

The work is a summary and extension of the master thesis of Dengiz, of which Mauser and this author were advisers. Mauser and this author aided Dengiz in developing the modeling of the optimization problem. Furthermore, Mauser identified relevant components of the optimization and developed a new taxonomy of characterizing optimizable household devices. This author carried out the analysis of the optimization results.

- Marlon Alexander Braun and Pradyumn Kumar Shukla. On cone based decompositions of proper Pareto optimality. *Optimization Online*, 2016.

M-domination and its cone-based decompositions were developed by this author. Shukla provided assistance in developing mathematical proofs that are related to both concepts.

- Marlon Alexander Braun, Sandra Seijo, Javier Echanobe, Pradyumn Kumar Shukla, Ines del Campo, Javier Garcia-Sedano, and Hartmut Schmeck. A neuro-genetic approach for modeling and optimizing a complex cogeneration process. *Applied Soft Computing*, 48:347 – 358, 2016.

Data collection, literature review and training of the neural networks was carried out by Seijo, Echanobe, del Campo and Garcia-Sedano. Optimization and analysis of the results was carried out by Braun. Shukla provided support in analyzing the results. Schmeck provided commentary and advice.

- Marlon Alexander Braun, Pradyumn Shukla, and Hartmut Schmeck. Angle-based preference models in multi-objective optimization. In *International Conference on Evolutionary Multi-Criterion Optimization*, pages 88–102. Springer, 2017.

Conceptual work and implementation was carried out by this author. Shukla and Schmeck provided commentary and advice.

- Marlon Alexander Braun, Lars Heling, Pradyumn Shukla, and Hartmut Schmeck. Multimodal scalarized preferences in multi-objective optimization. In *Proceedings of the 2017 on Genetic and Evolutionary Computation Conference*. ACM, 2017.

Conceptual work and analysis of the results was done by this author. Heling carried out the implementation and the experimental planning. Shukla and Schmeck provided commentary and advice.



## B. Problem Definitions

This appendix contains the mathematical descriptions of the benchmark problems used in the computational studies of Chapters 5 and 6.

**Definition 69** (B problem family). *Let  $n \geq 2$  and for any  $\mathbf{x} \in \mathbb{R}^n$*

$$g_1(\mathbf{x}) = \sum_{i=3}^n x_i \quad (\text{B.1a})$$

$$g_2(\mathbf{x}) = \sum_{i=3}^n x_i^2. \quad (\text{B.1b})$$

The B problem family is a set of MOOPs that is defined as follows:

- B1:

$$f_1(\mathbf{x}) = \sqrt{x_1^2 + x_2^2} \cdot (1 + g_1(\mathbf{x})) \quad (\text{B.2a})$$

$$f_2(\mathbf{x}) = \sqrt{x_1^2 + (1 - x_2)^2} \cdot (1 + g_1(\mathbf{x})) \quad (\text{B.2b})$$

$$f_3(\mathbf{x}) = (1 - x_1)(1 + g_1(\mathbf{x})) \quad (\text{B.2c})$$

$$\text{s.t. } \mathbf{x} \in [0, 1]^n, \quad (\text{B.2d})$$

- B2:

$$f_1(\mathbf{x}) = \sqrt{x_1^2 + x_2^2} \cdot (1 + g_1(\mathbf{x})) \quad (\text{B.3a})$$

$$f_2(\mathbf{x}) = \sqrt{1 + x_1^2 - x_2^2} \cdot (1 + g_1(\mathbf{x})) \quad (\text{B.3b})$$

$$f_3(\mathbf{x}) = \sqrt{1 - x_1^2} (1 + g_1(\mathbf{x})) \quad (\text{B.3c})$$

$$\text{s.t. } \mathbf{x} \in [0, 1]^n, \quad (\text{B.3d})$$

- B3:

$$f_1(\mathbf{x}) = (x_1^2 + x_2^2) (1 + g_1(\mathbf{x})) \quad (\text{B.4a})$$

$$f_2(\mathbf{x}) = (x_1^2 + (1 - x_2)^2) (1 + g_1(\mathbf{x})) \quad (\text{B.4b})$$

$$f_3(\mathbf{x}) = (1 - x_1)^2 (1 + g_1(\mathbf{x})) \quad (\text{B.4c})$$

$$\text{s.t. } \mathbf{x} \in [0, 1]^n, \quad (\text{B.4d})$$

## B. Problem Definitions

- B4:

$$f_1(\mathbf{x}) = (x_1^2 + x_2^2) (1 + g_1(\mathbf{x})) \quad (\text{B.5a})$$

$$f_2(\mathbf{x}) = (1 + x_1^2 - x_2^2) (1 + g_1(\mathbf{x})) \quad (\text{B.5b})$$

$$f_3(\mathbf{x}) = (1 - x_1^2) (1 + g_1(\mathbf{x})) \quad (\text{B.5c})$$

$$\text{s.t. } \mathbf{x} \in [0, 1]^n, \quad (\text{B.5d})$$

- B5:

$$f_1(\mathbf{x}) = -x_1(1 + g_2(\mathbf{x})) \quad (\text{B.6a})$$

$$f_2(\mathbf{x}) = -x_2^3(1 + g_2(\mathbf{x})) \quad (\text{B.6b})$$

$$f_3(\mathbf{x}) = (x_1 + x_2) (1 + g_2(\mathbf{x})) \quad (\text{B.6c})$$

$$\text{s.t. } \mathbf{x} \in [-1, 1]^n, \quad (\text{B.6d})$$

- B6:

$$f_1(\mathbf{x}) = -x_1^3(1 + g_2(\mathbf{x})) \quad (\text{B.7a})$$

$$f_2(\mathbf{x}) = -x_2^3(1 + g_2(\mathbf{x})) \quad (\text{B.7b})$$

$$f_3(\mathbf{x}) = (x_1 + x_2) (1 + g_2(\mathbf{x})) \quad (\text{B.7c})$$

$$\text{s.t. } \mathbf{x} \in [-1, 1]^n. \quad (\text{B.7d})$$

**Definition 70** (DEB2DK [BDDO04]). Let  $n \geq 1, k \in \mathbb{N}$  and for any  $\mathbf{x} \in \mathbb{R}^n$

$$g(\mathbf{x}) = 1 + \frac{9}{n-1} \sum_{i=2}^n x_i \quad (\text{B.8a})$$

$$r(\mathbf{x}) = 5 + 10 \left( x_1 - \frac{1}{2} \right)^2 + \frac{\cos(2k\pi x_1)}{K}. \quad (\text{B.8b})$$

DEB2DK is an MOOP defined as follows:

$$f_1(\mathbf{x}) = \sin\left(\frac{\pi}{2}x_1\right) g(\mathbf{x})r(\mathbf{x}) \quad (\text{B.9a})$$

$$f_2(\mathbf{x}) = \cos\left(\frac{\pi}{2}x_1\right) g(\mathbf{x})r(\mathbf{x}) \quad (\text{B.9b})$$

$$\text{s.t. } \mathbf{x} \in [0, 1]^n. \quad (\text{B.9c})$$

**Definition 71** (DEB3DK [BDDO04]). Let  $n \geq 1, k \in \mathbb{N}$  and for any  $\mathbf{x} \in \mathbb{R}^n$

$$g(\mathbf{x}) = 1 + \frac{9}{n-1} \sum_{i=3}^n x_i \quad (\text{B.10a})$$

$$r(x_1, x_2) = \frac{r_1(x_1) + r_2(x_2)}{2} \quad (\text{B.10b})$$

$$r_i(x_i) = 5 + 10 \left( x_i - \frac{1}{2} \right)^2 + \frac{\cos(2k\pi x_i)}{K}. \quad (\text{B.10c})$$

DE32DK is an MOOP defined as follows:

$$f_1(\mathbf{x}) = \sin\left(\frac{\pi}{2}x_1\right) \sin\left(\frac{\pi}{2}x_2\right) g(\mathbf{x})r(x_1, x_2) \quad (\text{B.11a})$$

$$f_2(\mathbf{x}) = \cos\left(\frac{\pi}{2}x_1\right) \cos\left(\frac{\pi}{2}x_2\right) g(\mathbf{x})r(x_1, x_2) \quad (\text{B.11b})$$

$$f_3(\mathbf{x}) = \cos\left(\frac{\pi}{2}x_1\right) g(\mathbf{x})r(x_1, x_2) \quad (\text{B.11c})$$

$$\text{s.t. } \mathbf{x} \in [0, 1]^n. \quad (\text{B.11d})$$

Note that the original description of DEB3DK in [BDDO04] contains an error. In Equation (B.10b) the index of the sum starts at  $i = 2$  instead of  $i = 3$ . The index starting at  $i = 2$  results in the Pareto front of DEB3DK being asymmetric, which would contradict the illustration in [BDDO04, Fig. 6].

**Definition 72** (DO2DK [BDDO04]). Let  $n \geq 1$ ,  $k \in \mathbb{N}$ ,  $s \in \mathbb{R}_+$  and for any  $\mathbf{x} \in \mathbb{R}^n$

$$g(\mathbf{x}) = 1 + \frac{9}{n-1} \sum_{i=2}^n x_i \quad (\text{B.12a})$$

$$r(\mathbf{x}) = 5 + 10 \left( x_1 - \frac{1}{2} \right)^2 + \frac{2^{\frac{s}{2}} \cos(2k\pi x_1)}{K}. \quad (\text{B.12b})$$

DEB2DK is an MOOP defined as follows:

$$f_1(\mathbf{x}) = \sin\left(\frac{\pi}{2}x_1 + \left(1 + \frac{2^s - 1}{2^{s+2}}\right) \pi + 1\right) g(\mathbf{x})r(\mathbf{x}) \quad (\text{B.13a})$$

$$f_2(\mathbf{x}) = \left(\cos\left(\frac{\pi}{2}x_1 + \pi\right) + 1\right) g(\mathbf{x})r(\mathbf{x}) \quad (\text{B.13b})$$

$$\text{s.t. } \mathbf{x} \in [0, 1]^n. \quad (\text{B.13c})$$

**Definition 73** (DTLZ problem definitions [DTLZ05, JD14]). Let  $n \geq 2$ . The DTLZ problem family is a set of MOOPs that is defined as follows:

- DTLZ1 ([DTLZ05])

$$f_1(\mathbf{x}) = \frac{1}{2} x_1 \cdots x_{m-1} (1 + g(\mathbf{x})) \quad (\text{B.14a})$$

$$f_2(\mathbf{x}) = \frac{1}{2} x_1 \cdots x_{m-2} (1 - x_{m-1}) (1 + g(\mathbf{x})) \quad (\text{B.14b})$$

## B. Problem Definitions

$$f_3(\mathbf{x}) = \frac{1}{2}x_1 \cdots x_{m-3}(1 - x_{m-2})(1 + g(\mathbf{x})) \quad (\text{B.14c})$$

$$\vdots = \vdots \quad (\text{B.14d})$$

$$f_{m-1}(\mathbf{x}) = \frac{1}{2}x_1(1 - x_2)(1 + g(\mathbf{x})) \quad (\text{B.14e})$$

$$f_m(\mathbf{x}) = \frac{1}{2}(1 - x_1)(1 + g(\mathbf{x})) \quad (\text{B.14f})$$

$$g(\mathbf{x}) = 100 \left( n - m + 1 + \sum_{i=m}^n \left( x_i - \frac{1}{2} \right)^2 - \cos \left( 20\pi \left( x_i - \frac{1}{2} \right) \right) \right) \quad (\text{B.14g})$$

$$\text{s.t. } \mathbf{x} \in [0, 1]^n, \quad (\text{B.14h})$$

- *inverted DTLZ1 ([JD14])*

$$f_1(\mathbf{x}) = (1 - x_1 \cdots x_{m-1}) \frac{1}{2}(1 + g(\mathbf{x})) \quad (\text{B.15a})$$

$$f_2(\mathbf{x}) = (1 - x_1 \cdots x_{m-2}(1 - x_{m-1})) \frac{1}{2}(1 + g(\mathbf{x})) \quad (\text{B.15b})$$

$$f_3(\mathbf{x}) = (1 - x_1 \cdots x_{m-3}(1 - x_{m-2})) \frac{1}{2}(1 + g(\mathbf{x})) \quad (\text{B.15c})$$

$$\vdots = \vdots \quad (\text{B.15d})$$

$$f_{m-1}(\mathbf{x}) = (1 - x_1(1 - x_2)) \frac{1}{2}(1 + g(\mathbf{x})) \quad (\text{B.15e})$$

$$f_m(\mathbf{x}) = (1 - (1 - x_1)) \frac{1}{2}(1 + g(\mathbf{x})) \quad (\text{B.15f})$$

$$g(\mathbf{x}) = 100 \left( n - m + 1 + \sum_{i=m}^n \left( x_i - \frac{1}{2} \right)^2 - \cos \left( 20\pi \left( x_i - \frac{1}{2} \right) \right) \right) \quad (\text{B.15g})$$

$$\text{s.t. } \mathbf{x} \in [0, 1]^n, \quad (\text{B.15h})$$

- *DTLZ3 ([DTLZ05])*

$$f_1(\mathbf{x}) = \cos \left( \frac{\pi}{2} x_1 \right) \cdots \cos \left( \frac{\pi}{2} x_{m-1} \right) (1 + g(\mathbf{x})) \quad (\text{B.16a})$$

$$f_2(\mathbf{x}) = \cos \left( \frac{\pi}{2} x_1 \right) \cdots \cos \left( \frac{\pi}{2} x_{m-2} \right) \sin \left( \frac{\pi}{2} x_{m-2} \right) (1 + g(\mathbf{x})) \quad (\text{B.16b})$$

$$f_3(\mathbf{x}) = \cos \left( \frac{\pi}{2} x_1 \right) \cdots \cos \left( \frac{\pi}{2} x_{m-3} \right) \sin \left( \frac{\pi}{2} x_{m-2} \right) (1 + g(\mathbf{x})) \quad (\text{B.16c})$$

$$\vdots = \vdots \quad (\text{B.16d})$$

$$f_{m-1}(\mathbf{x}) = \cos\left(\frac{\pi}{2}x_1\right) \sin\left(\frac{\pi}{2}x_2\right) (1 + g(\mathbf{x})) \quad (\text{B.16e})$$

$$f_m(\mathbf{x}) = \sin\left(\frac{\pi}{2}x_1\right) (1 + g(\mathbf{x})) \quad (\text{B.16f})$$

$$g(\mathbf{x}) = 100 \left( n - m + 1 + \sum_{i=m}^n \left( x_i - \frac{1}{2} \right)^2 - \cos\left( 20\pi \left( x_i - \frac{1}{2} \right) \right) \right) \quad (\text{B.16g})$$

$$\text{s.t. } \mathbf{x} \in [0, 1]^n, \quad (\text{B.16h})$$

- *DTLZ5 ([DTLZ05])*

$$f_1(\mathbf{x}) = \cos\left(\frac{\pi}{2}h(x_1)\right) \cdots \cos\left(\frac{\pi}{2}h(x_{m-1})\right) (1 + g(\mathbf{x})) \quad (\text{B.17a})$$

$$f_2(\mathbf{x}) = \cos\left(\frac{\pi}{2}h(x_1)\right) \cdots \cos\left(\frac{\pi}{2}h(x_{m-2})\right) \sin\left(\frac{\pi}{2}h(x_{m-1})\right) (1 + g(\mathbf{x})) \quad (\text{B.17b})$$

$$f_3(\mathbf{x}) = \cos\left(\frac{\pi}{2}h(x_1)\right) \cdots \cos\left(\frac{\pi}{2}h(x_{m-3})\right) \sin\left(\frac{\pi}{2}h(x_{m-2})\right) (1 + g(\mathbf{x})) \quad (\text{B.17c})$$

$$\vdots = \vdots \quad (\text{B.17d})$$

$$f_{m-1}(\mathbf{x}) = \cos\left(\frac{\pi}{2}h(x_1)\right) \sin\left(\frac{\pi}{2}h(x_2)\right) (1 + g(\mathbf{x})) \quad (\text{B.17e})$$

$$f_m(\mathbf{x}) = \sin\left(\frac{\pi}{2}h(x_1)\right) (1 + g(\mathbf{x})) \quad (\text{B.17f})$$

$$h(x_i) = \frac{\pi}{4(1 + g(\mathbf{x}))} (1 + 2g(\mathbf{x})x_i) \quad (\text{B.17g})$$

$$g(\mathbf{x}) = \sum_{i=m}^n \left( x_i - \frac{1}{2} \right)^2 \quad (\text{B.17h})$$

$$\text{s.t. } \mathbf{x} \in [0, 1]^n, \quad (\text{B.17i})$$

- *DTLZ7 ([DTLZ05])*

$$f_1(\mathbf{x}) = x_1 \quad (\text{B.18a})$$

$$\vdots = \vdots \quad (\text{B.18b})$$

$$f_{m-1}(\mathbf{x}) = x_{m-1} \quad (\text{B.18c})$$

$$f_m(\mathbf{x}) = m - \sum_{i=1}^{m-1} \left( \frac{x_i}{1 + g(\mathbf{x})} (1 + \sin(3\pi x_i)) \right) \quad (\text{B.18d})$$

$$g(\mathbf{x}) = 1 + \frac{9}{n - m + 1} \sum_{i=m}^n x_i \quad (\text{B.18e})$$

## B. Problem Definitions

$$\text{s.t. } \mathbf{x} \in [0, 1]^n, \quad (\text{B.18f})$$

Note that the DTLZ problem family consists of nine MOOPs in total of which only four are considered in this work.

**Definition 74** (Lamé [ED07]). *Let  $n \geq 1$ ,  $\gamma \in \mathbb{R}^+$  and for any  $\mathbf{x} \in \mathbb{R}^n$*

$$g(\mathbf{x}) = \sqrt{\sum_{i=m}^n x_i^2}. \quad (\text{B.19})$$

The Lamé problem is an MOOP defined as follows:

$$f_1(\mathbf{x}) = \left( \cos\left(\frac{\pi}{2}x_1\right) \right)^{\frac{2}{\gamma}} (1 + g(\mathbf{x})) \quad (\text{B.20a})$$

$$f_2(\mathbf{x}) = \left( \sin\left(\frac{\pi}{2}x_1\right) \cos\left(\frac{\pi}{2}x_2\right) \right)^{\frac{2}{\gamma}} (1 + g(\mathbf{x})) \quad (\text{B.20b})$$

$$f_3(\mathbf{x}) = \left( \sin\left(\frac{\pi}{2}x_1\right) \sin\left(\frac{\pi}{2}x_2\right) \cos\left(\frac{\pi}{2}x_3\right) \right)^{\frac{2}{\gamma}} (1 + g(\mathbf{x})) \quad (\text{B.20c})$$

$$\vdots = \vdots \quad (\text{B.20d})$$

$$f_{m-1}(\mathbf{x}) = \left( \sin\left(\frac{\pi}{2}x_1\right) \cdots \sin\left(\frac{\pi}{2}x_{m-2}\right) \cos\left(\frac{\pi}{2}x_{m-1}\right) \right)^{\frac{2}{\gamma}} (1 + g(\mathbf{x})) \quad (\text{B.20e})$$

$$f_m(\mathbf{x}) = \left( \sin\left(\frac{\pi}{2}x_1\right) \cdots \sin\left(\frac{\pi}{2}x_{m-2}\right) \sin\left(\frac{\pi}{2}x_{m-1}\right) \right)^{\frac{2}{\gamma}} (1 + g(\mathbf{x})) \quad (\text{B.20f})$$

$$\text{s.t. } \mathbf{x} \in [0, 1]^n. \quad (\text{B.20g})$$

**Definition 75** (ZDT problem family [ZDT00]). *Let  $n \geq 2$  and for any  $\mathbf{x} \in \mathbb{R}^n$*

$$g(\mathbf{x}) = 1 + 9 \sum_{i=2}^n \frac{x_i}{n-1}. \quad (\text{B.21})$$

The ZDT problem family is a set of MOOPs that is defined as follows:

- ZDT1:

$$f_1(\mathbf{x}) = x_1 \quad (\text{B.22a})$$

$$f_2(\mathbf{x}) = 1 - \sqrt{\frac{x_1}{g(\mathbf{x})}} \quad (\text{B.22b})$$

$$\text{s.t. } \mathbf{x} \in [0, 1]^n, \quad (\text{B.22c})$$

- ZDT2:

$$f_1(\mathbf{x}) = x_1 \quad (\text{B.23a})$$

$$f_2(\mathbf{x}) = 1 - \left( \frac{x_1}{g(\mathbf{x})} \right)^2 \quad (\text{B.23b})$$

$$s.t. \mathbf{x} \in [0, 1]^n, \quad (\text{B.23c})$$

- ZDT3:

$$f_1(\mathbf{x}) = x_1 \quad (\text{B.24a})$$

$$f_2(\mathbf{x}) = 1 - \sqrt{\frac{x_1}{g(\mathbf{x})}} - \frac{x_1}{g(\mathbf{x})} \sin(10\pi x_1) \quad (\text{B.24b})$$

$$s.t. \mathbf{x} \in [0, 1]^n. \quad (\text{B.24c})$$

Note that the ZDT problem family consists of six MOOPs in total of which only three are considered in this work.



## C. Algorithms

This appendix contains subroutines that are invoked in the algorithms described in Chapters 5 and 6. The subroutines are discussed in prose in the respective chapters.

---

**Algorithm 18:** update swarm [NDGN<sup>+</sup>09]

---

**Input :** Swarm  $S := (\mathbf{x}^1, \dots, \mathbf{x}^N)$ , velocities  $Z := (\mathbf{z}^1, \dots, \mathbf{z}^N)$ , global best archive  $A$ , personal best archive  $B$ , inertia weight  $w$ , turbulence factors  $c_1, c_2$

**Output:** Updated swarm  $S$  and velocities  $Z$

```

1 if  $c_1 + c_2 > 4$  then
2   |  $\alpha := c_1 + c_2$ 
3 else
4   |  $\alpha := 1$ 
5    $\chi := \frac{2}{2 - \alpha - \sqrt{\alpha^2 - 4\alpha}}$ 
6 for  $i = 1$  to  $N$  do
7   |  $r_1 := \mathcal{U}(0, 1)$ 
8   |  $r_2 := \mathcal{U}(0, 1)$ 
9   |  $\mathbf{z}^i := w\mathbf{z}^i + c_1r_1(B(i) - \mathbf{x}^i) + c_2r_2(\text{random}(A) - \mathbf{x})$ 
10  |  $\mathbf{z}^i := \chi\mathbf{z}^i$ 
11  |  $\mathbf{x}^i := \mathbf{x}^i + \mathbf{z}^i$ 
12 return  $(S, Z)$ 

```

---



---

**Algorithm 19:** hypervolume contributions [BNE07]

---

**Input :** Population  $P := (\mathbf{x}^1, \dots, \mathbf{x}^{|P|})$ , reference point  $\mathbf{w}$

**Output:** Hypervolume contributions  $\mathbf{c}$

```

1 Let  $L(P, \mathbf{w}) := \mathcal{L}(\bigcup_{\mathbf{x}^i \in P} (\mathbf{u} \in \mathbb{R}^m \mid \mathbf{f}^i <_p \mathbf{u} <_p \mathbf{w}))$ 
2 Compute  $\mathbf{c}$  with  $c_i := L(P, \mathbf{w}) - L(P \setminus \{\mathbf{x}^i\}, \mathbf{w})$ 
3 return  $\mathbf{c}$ 

```

---

### C. Algorithms

---

**Algorithm 20:** update mean [Han06]

---

**Input** : population  $P := \{\mathbf{x}^i\}_{i=1}^N$ , weights  $\lambda$ , parameter  $\mu$

**Output:** Updated mean  $\mathbf{y}$

- 1  $\mathbf{y} := \sum_{i=1}^{\mu} \lambda_i x_i$
  - 2 **return**  $\mathbf{y}$
- 

---

**Algorithm 21:** update isotropic evolution path [Han06]

---

**Input** : Isotropic evolution path  $\mathbf{p}_\sigma$ , parameters  $c_\sigma, \mu_w$ , covariance matrix  $\mathbf{V}$ , old distribution mean  $\mathbf{y}^{old}$ , new distribution mean  $\mathbf{y}$ , step size  $\sigma$

**Output:** Updated isotropic evolution path  $\mathbf{p}_\sigma$

- 1  $\mathbf{p}_\sigma := (1 - c_\sigma)\mathbf{p}_\sigma + \sqrt{1 - (1 - c_\sigma)^2} \sqrt{\mu_w \mathbf{V}^{-1} \frac{\mathbf{y} - \mathbf{y}^{old}}{\sigma}}$
  - 2 **return**  $\mathbf{p}_\sigma$
- 

---

**Algorithm 22:** update anisotropic evolution path [Han06]

---

**Input** : Anisotropic evolution path  $\mathbf{p}_c$ , parameters  $c_c, \alpha, \mu_w$ , isotropic evolution path  $\mathbf{p}_\sigma$ , old distribution mean  $\mathbf{y}^{old}$ , new distribution mean  $\mathbf{y}$ , step size  $\sigma$

**Output:** Updated anisotropic evolution path

- 1  $\mathbf{p}_c := (1 - c_c)\mathbf{p}_c + 1_{[0, \alpha]} (\|\mathbf{p}_\sigma\|_2) \sqrt{1 - (1 - c_c)^2} \sqrt{\mu_w \frac{\mathbf{y}^{old} - \mathbf{y}}{\sigma}}$
  - 2 **return**  $\mathbf{p}_c$
- 

---

**Algorithm 23:** update covariance matrix [Han06]

---

**Input** : Covariance matrix  $\mathbf{V}$ , parameters  $c_1, c_\mu, c_s$ , anisotropic evolution path  $\mathbf{p}_\sigma$ , weights  $\lambda$ , population  $P$ , old distribution mean  $\mathbf{y}^{old}$ , step size  $\sigma$

**Output:** Update covariance matrix  $\mathbf{V}$

- 1  $\mathbf{V} := (1 - c_1 - c_\mu + c_s)\mathbf{V} + c_1 \mathbf{p}_c \mathbf{p}_c^T + c_\mu \sum_{i=1}^{\mu} \lambda_i \frac{\mathbf{z}^i - \mathbf{y}}{\sigma} \left( \frac{\mathbf{z}^i - \mathbf{y}}{\sigma} \right)^T$
  - 2 **return**  $\mathbf{V}$
- 

---

**Algorithm 24:** update step size [Han06]

---

**Input** : step size  $\sigma$ , parameters  $c_\sigma, d_\sigma$  isotropic evolution path  $\mathbf{p}_\sigma$

**Output:** updated step size  $\sigma$

- 1  $\sigma := \sigma \cdot \exp\left(\frac{c_\sigma}{d_\sigma} \left(\frac{\|\mathbf{p}_\sigma\|}{\mathbb{E}(\|\mathcal{N}_m(0, I)\|)} - 1\right)\right)$
  - 2 **return**  $\sigma$
-

## D. Basin Construction

The method that was used to find local scalarization optima of three objective problems and their corresponding basins of attraction is described in this appendix. As explained in Section 5.3.1, the three objective artificial benchmark problems considered in this work are non-conflicting in the decision variables  $\{x_3, \dots, x_n\}$ . This implies that there exists a vector  $\mathbf{a}$  such that  $(x_3, \dots, x_n) = \mathbf{a}$  minimizes the objective functions. Let  $\mathbf{x}^l$  and  $\mathbf{x}^u$  denote the lower and upper bounds of the feasible set  $X$  of  $f$ . The Pareto optimal set can be expressed as

$$X_p := \{\mathbf{x} \in X \mid \forall i \in \{1, 2\} : \mathbf{x}_i^l \leq x_i \leq \mathbf{x}_i^u \text{ and } (x_3, \dots, x_n) = \mathbf{a}\}. \quad (\text{D.1})$$

The basic idea of the basin construction approach consists of discretizing the bounding box of  $(x_1, x_2)$  into a grid such that it forms a lattice graph. Nodes that are adjacent in  $G$  are also adjacent in the objective space. The algorithm then builds a directed graph  $G$  in a similar manner as done in TS and NBC. Let  $a_{i,j}$  denote a node in the lattice graph at position  $(i, j)$  in the grid. The node  $a_{i,j}$  is compared to all nodes  $a_{k,l}$  in every cardinal and intermediate cardinal direction. If an  $a_{k,l}$  possesses a smaller scalarization value than  $a_{i,j}$ , an edge pointing from  $a_{i,j}$  to  $a_{k,l}$  is added to  $G$ . Any node that possesses no outgoing edges is an estimate of a local scalarization optimum. Any path in  $G$  from  $a_{i,j}$  to a local optimum represents a descending search path in the scalarization space (see function  $\alpha$  in Definition 60).

If  $G$  consists of disconnected subgraphs, these form the basins of  $f$  for a given scalarization function  $\Psi$ . Depending on the topology of the scalarization landscape, it may happen that multiple local optima are reachable from a  $a_{i,j}$ . If this is the case  $a_{i,j}$  is assigned to the basin of the closest local optimum. The distance is defined as the minimum number of nodes that need to be traversed for traveling from  $a_{i,j}$  to the local minimum.

In order to make the results of the algorithm more robust, a parameter  $t$  was introduced that defines the minimum neighborhood, in which a node  $a_{i,j}$  has to have the smallest scalarization value to qualify as local optimum. The neighborhood consists of all those nodes that lie within  $t$  steps in horizontal or vertical direction of  $a_{i,j}$ . The parameter  $t$  can be interpreted as minimum size for a subgraph to qualify as basin.

#### D. Basin Construction

Note that there exist problems for which the set  $X_p$  as given in Equation (D.1) also contains Pareto-dominated solutions. This is for example the case for DEB3DK. For this reason, Pareto-dominated elements of the lattice graph are ignored in constructing the graph  $G$ . The complete procedure is summarized in Algorithm 25. The resolution of the grid, i.e. the number of nodes in each dimension, is controlled by the parameter  $s$ .

---

#### Algorithm 25: basin construction algorithm

---

**Input** : MOOP  $f$ , scalarization function  $\Psi$ , parameters  $s, t$   
**Output**: Local optima  $\mathcal{L}$ , basin system  $\mathcal{B}$

- 1  $\mathcal{L} := (), \mathcal{B} := ()$
- 2 Let  $\mathbf{x}^l$  and  $\mathbf{x}^u$  be the lower and upper bounds of the feasible set of  $f$
- 3 Let  $a_{i,j} := (x_1^l + (i-1)(x_1^u - x_1^l)/s, x_2^l + (j-1)(x_2^u - x_2^l)/s)$
- 4 Create directed graph  $G := (V, E)$  with  $V := \{a_{i,j} \mid a_{i,j} \in X_p\}$  and  $E := \emptyset$
- 5 **forall**  $a_{i,j}$  with  $i, j \in [s+1]$  **do**
- 6     **if**  $a_{i,j} \in X_p$  **then**
- 7         // find local optima
- 7         **if**  $\forall k \in [i-t, i+t]$  and  $l \in [j-t, j+t]$  with  $a_{k,l} \in X_p$ :  
         $f(a_{k,l}) \not\prec_p f(a_{i,j})$  and  $\Psi(a_{i,j}) \leq \Psi(a_{k,l})$  **then**
- 8              $\mathcal{L} := (\mathcal{L}, \{a_{i,j}\})$
- 9         // build directed graph
- 9         **forall**  $a_{k,l} \in X_p$  with  $k \in [i-1, i+1]$  and  $l \in [j-1, j+1]$  **do**
- 10             **if**  $\Psi(f(a_{k,l})) \leq \Psi(f(a_{i,j}))$  **then**
- 11                  $E := E \cup (a_{i,j}, a_{k,l})$
- // construct basins
- 12  $\mathcal{B} := (\{\mathcal{L}(1)\}, \dots, \{\mathcal{L}(|\mathcal{L}|)\})$
- 13 **forall**  $a_{i,j} \in G$  **do**
- 14      $k = \arg \min_{l \in [|\mathcal{L}|]} (d(a_{i,j}, \mathcal{L}(l)))$
- 15      $\mathcal{B}(k) := \mathcal{B}(k) \cup a_{i,j}$
- 16 **return**  $(\mathcal{L}, \mathcal{B})$

---

## E. Experimental Settings

This appendix summarizes the experimental settings for the computational studies reported throughout this work. The setup depicted in Table E.1 was used in the study described in Section 5.3.1. The values listed in Table E.2 were utilized for the studies in Sections 5.3.2 and 7.1.2. The setup of ESPEA contained in Table E.2 was also used in the studies in Section 5.3.3 and Section 6.3.

Table E.1.: Experimental settings in the replacement strategies study.

replacement strategies	{BFP, LED, WIN}
performance indicators	{RE, IGD}
archive size	$N = 50$
function evaluations	50 000
number of runs	100
sampling of initial population	
RE study	uniform
IGD study	latin hypercube
normalized objectives for Euclidean distance	
RE study	false
IGD study	true
mating selection	random selection
crossover	
archive not full	SBX
distribution index	$\eta_c = 20$
recombination probability	$p_c = 1$
archive full	DE
reproduction scheme	<i>current-to-rand/1/bin</i>
coefficient $c_k$	$c_k = 0.5$
coefficient $c_f$	$c_f = 0.5$
recombination probability	$p_c = 0.5$
mutation	polynomial mutation
distribution index	$\eta_m = 20$
mutation probability	$1/m$

## E. Experimental Settings

Table E.2.: Experimental settings in the comparative study of ESPEA and the cogeneration study.

performance indicators	{IGD}
population/archive size	$N = 100$
function evaluations	50 000
number of runs	100
sampling of initial population	latin hypercube
crossover	
SBX	
recombination probability	$p_c = 1$
DE	
coefficient $c_k$	$c_k = 0.5$
coefficient $c_f$	$c_f = 0.5$
recombination probability	$p_c = 0.5$
mutation	polynomial mutation
distribution index	$\eta_m = 20$
mutation probability	$1/m$
<hr/>	
ESPEA	
mating selection	random selection
replacement strategy	WIN
crossover	
archive not full	SBX with $\eta_c = 20$
archive full	DE with <i>current-to-rand/1/bin</i>
mutation	polynomial mutation
<hr/>	
IBEA	
scaling factor	$\kappa = 0.05$
mating selection	binary tournament
selection criterion	fitness
crossover	SBX with $\eta_c = 20$
mutation	polynomial mutation
<hr/>	
MOEAD	
neighborhood size	20
neighborhood selection probability	$p_b = 0.9$
max neighbors to replace	$c_N = 2$
reference points	unit simplex ([DD98, Sec. 5])
crossover	DE with <i>rand/1/bin</i>
mutation	polynomial mutation
NSGA-II	

mating selection	binary tournament
selection criterion	rank and crowding distance
crossover	SBX with $\eta_c = 20$
mutation	polynomial mutation
<hr/>	
NSGA-III	
mating selection	binary tournament
selection criterion	Pareto dominance
reference points	unit simplex ([DD98, Sec. 5])
crossover	SBX with $\eta_c = 30$
mutation	polynomial mutation
<hr/>	
SMPSO	inertia weight
turbulence factor	range
mutation	polynomial mutation
<hr/>	
SMS-EMOA	mating selection
selection criterion	hypervolume contributions
reference point	$10^m$
crossover	SBX with $\eta_c = 20$
mutation	polynomial mutation
<hr/>	
SPEA2	
nearest neighbor	$k = 1$
mating selection	binary tournament
selection criterion	fitness
crossover	SBX with $\eta_c = 30$
mutation	polynomial mutation

Table E.3.: ESPEA configuration in the multimodal preference study.

archive size	200
function evaluations	25 000
number of runs	100
replacement strategy	WIN
sampling of initial population	latin hypercube
mating selection	random selection
crossover	
archive not full	SBX
distribution index	$\eta_c = 20$
recombination probability	$p_c = 1$

## E. Experimental Settings

archive full	DE
reproduction scheme	<i>current-to-rand/1/bin</i>
coefficient $c_k$	$c_k = 0.5$
coefficient $c_f$	$c_f = 0.5$
recombination probability	$p_c = 0.5$
mutation	polynomial mutation
distribution index	$\eta_m = 20$
mutation probability	$1/m$

---

Table E.4.: Experimental settings in the basin identification study.

performance indicators	{UCF, DBF}
Pareto front approximation	ESPEA (see Table E.3)
number of runs	100
TS	$k_c \in \{2, 4, 6, 10, 20\}$
NBC	$\phi^c \in \{1.5, 2, 2.5, 3, 3.5\}$
$b_c$	see Equation (6.1)

---

Table E.5.: Experimental settings in the local search study.

performance indicators	{PD, PR, FP}
Pareto front approximation	ESPEA (see Table E.3)
basin identification	TS with $k_c = 6$
number of runs	100
CMA-ES	$\sigma \in \{1e-5, 1e-4, 1e-3, 1e-2, 1e-1\}$
GA	$\eta_m, \eta_c \in \{20, 30, 40, 80, 120\}$
recombination	SBX with $p_c = 1$
mutation	polynomial mutation with $p_m = 1/m$
HC	$\eta_m \in \{20, 30, 40, 80, 120\}$
mutation	polynomial mutation with $p_m = 1/m$
PSA	$w \in \{0.1, 0.25, 0.5, 0.75, 1\}$
turbulence factors	$c_1 = c_2 = 1$
constriction factor	$\chi = 1$

---

Table E.6.: Overview of CMA-ES strategy parameters and their chosen values.

---

$\alpha$	$1.4 + \frac{2}{n+1}$
$c_c$	$\frac{4 + \mu_w/n}{n + 4 + 2\mu_w/n}$
$c_s$	$\frac{\mu_w + 2}{n + \mu_w + 5}$
$c_1$	$\frac{2}{((n + 1.3)^2 + \mu_w)}$
$c_\mu$	$\min\left(1 - c_1, \frac{2(\mu_w - 2 + 1/\mu_w)}{(n + 2)^2 + \mu_w}\right)$
$d_\sigma$	$1 + c_\sigma + 2 \cdot \max\left(0, \sqrt{\frac{\mu_w - 1}{n + 1}} - 1\right)$
$\lambda$	$\lambda_i = \frac{\lambda'_i}{\sum_{i=1}^{\mu} \lambda'_i}$ with $\lambda'_i = \log\left(\frac{\mu + 1}{2(i + 1)}\right)$
$\mu$	$N/2$

---



## F. BEMS Study Documentation

This appendix contains the extended documentation of the BEMS study. A technical documentation of the devices used in the study is provided and an illustration of the time series data, i.e. load profiles, price curves, temperature curves, solar irradiance, electricity and hot water consumption. Note that these data are also contained in the digital appendix.

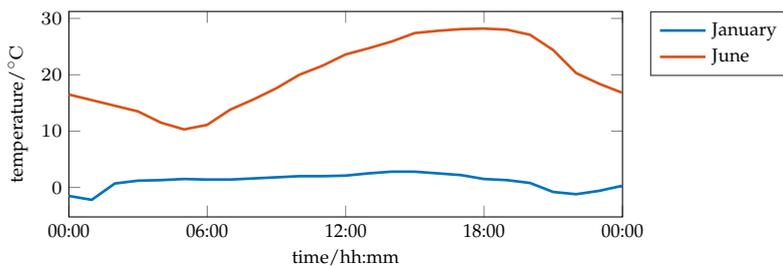


Figure F.1.: Profiles of outside temperature. Data obtained from *wetter.com*: 7.1.2015 and 4.6.2015, both Karlsruhe.

## F. BEMS Study Documentation

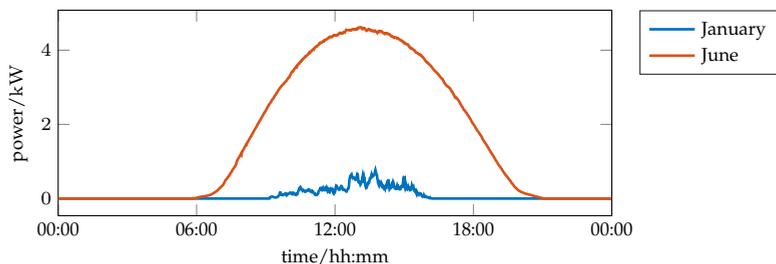


Figure F.2.: Solar irradiance. Data obtained from measurements the *ESHL*.

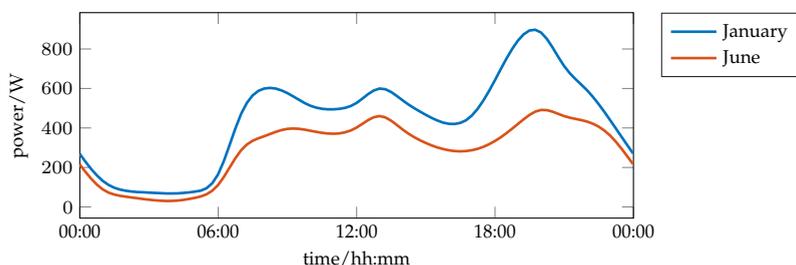


Figure F.3.: Residential building base load. Data is based on the *H0* load profile provided by the *German Association of Energy and Water Industries*. The load of optimizable devices was subtracted.

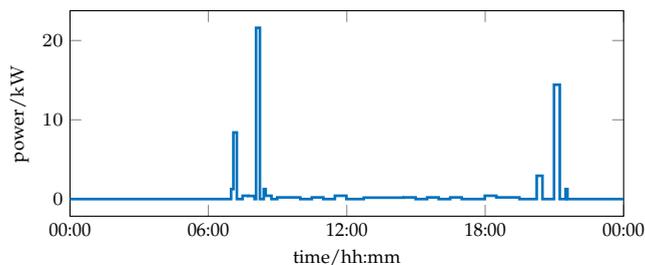


Figure F.4.: Hot water consumption profile of a residential building. Data obtained from *Directive 2010/30/EU* of the European Commission.

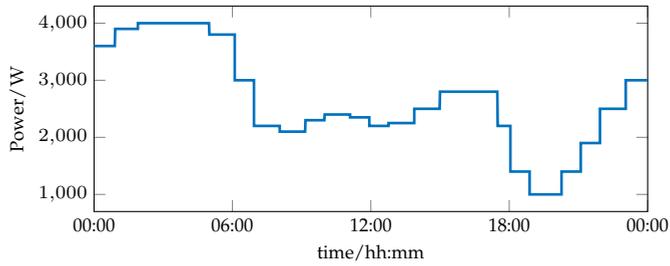


Figure F.5.: Load limitation signal above which the household has to pay a penalty for electricity consumption. Data obtained from [All14].

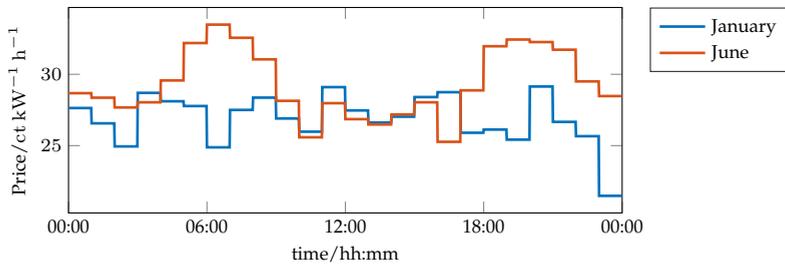


Figure F.6.: Price in cent per kilowatt hour of electricity purchased from the grid. Data obtained from [Dal13].

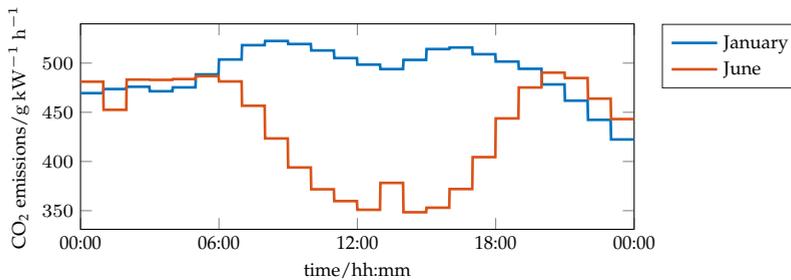


Figure F.7.: Carbon dioxide emissions of consuming a unit of electricity from the grid in January and June. Data obtained from *Fraunhofer Institute of Solar Energy*.

F. BEMS Study Documentation

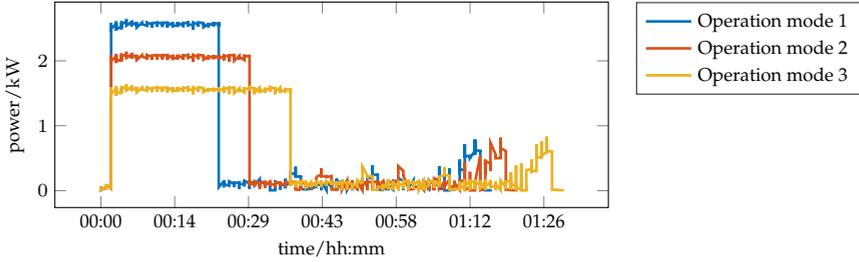


Figure F.8.: Load profiles of the different operation modes of the washing machine. Operation mode 2 was chosen as load profile of the deferrable washing machine. Data obtained from measurements in the *ESHL*.

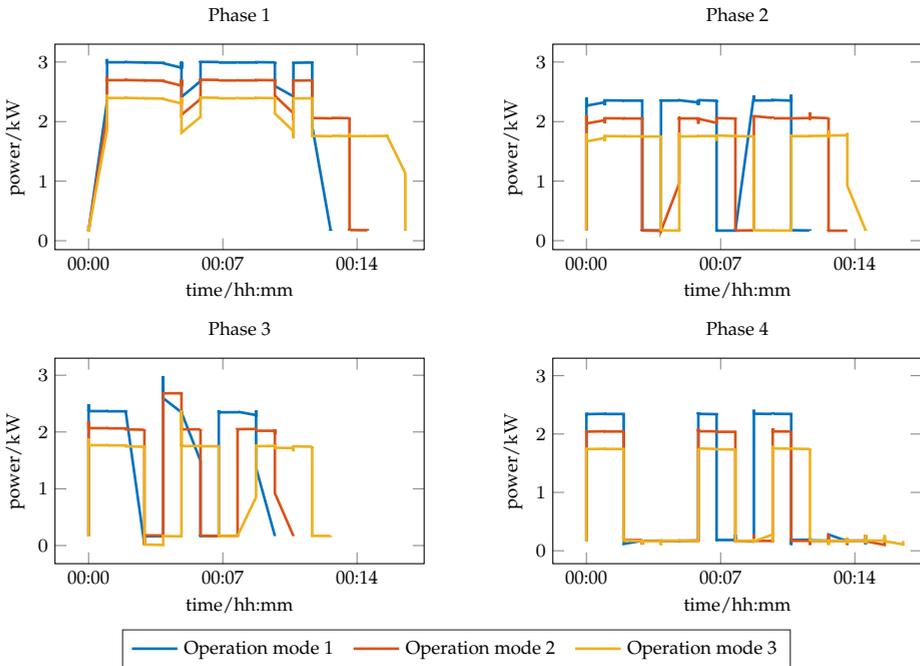


Figure F.9.: Load profiles of the different operation modes and phases of the tumble dryer. Operation mode 2 was chosen as load profile of the deferrable washing machine. Data obtained from measurements in the *ESHL*.

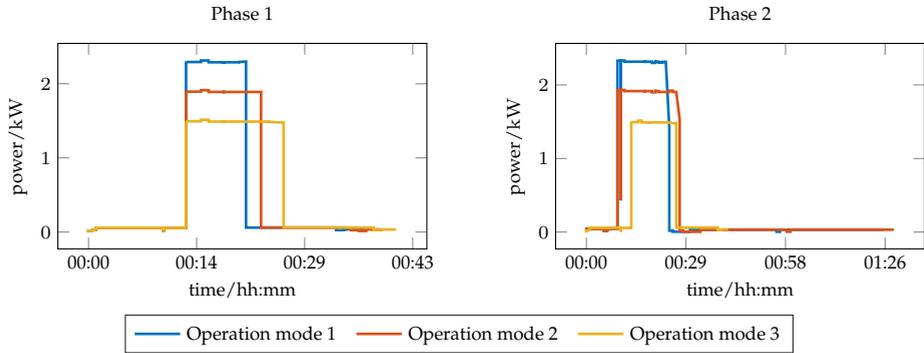


Figure F.10.: Load profiles of the different operation modes and phases of the dishwasher. Operation mode 2 was chosen as load profile of the deferrable and interruptible dishwasher. Data obtained from measurements in the *ESHL*.

Table F.1.: General parameter overview of the BEMS problem.

$A$	130 m <sup>2</sup>
$h$	2 m
$c^{\text{air}}$	1005 J kg <sup>-1</sup> K <sup>-1</sup>
$c^{\text{water}}$	4181 J kg <sup>-1</sup> K <sup>-1</sup>
$e^{\mathcal{S}}$	247 g kW <sup>-1</sup> h <sup>-1</sup>
$p^{\mathcal{S}}(t)$	9.16 ct kW <sup>-1</sup> h <sup>-1</sup>
$p_{\text{CHP}}$	12.56 ct kW <sup>-1</sup> h <sup>-1</sup>
$p_{\text{PV}}$	8.53 ct kW <sup>-1</sup> h <sup>-1</sup>
$Q_{\text{WS}}^{\text{max}}$	10 000 J
$Q^{\text{other}}(t)$	200 J for all $t$
$\rho^{\text{air}}$	1.204 kg m <sup>-3</sup>
$\rho^{\text{water}}$	988 kg m <sup>-3</sup>
$T_{\text{WS}}^{\text{heat}}$	50°
$T^{\text{l}}$	19.5° (winter), 21.5 (summer)
$T^{\text{max}}$	25°
$T_{\text{WS}}^{\text{max}}$	90°
$T^{\text{min}}$	18°
$T_{\text{WS}}^{\text{min}}$	40°
$T^{\text{set}}$	20° (winter), 22° (summer)
$T^{\text{u}}$	20.5° (winter), 22.5 (summer)
$V_{\text{WS}}$	0.4 m <sup>3</sup>

## F. BEMS Study Documentation

Table F.2.: Overview of appliances in summer and winter scenarios in the BEMS study.

	RW1/RS1	RW2/RW2	RW3/RS3
Washing machine	deferrable	deferrable	deferrable and load-flexible
Tumble dryer	deferrable	interruptible	interruptible and load-flexible
Dishwasher	deferrable	interruptible	interruptible and load-flexible
CHP plant			available
Condensing boiler			available
Air-conditioning			available
PV system		available only in summer	

Table F.3.: Temporal degree of freedom and standby consumptions of devices. The variable  $E_j^{\text{standby,init}}$  denotes the electricity consumption of device  $j$  before its first phase has been executed.

	Release	Deadline	$E_j^{\text{standby,init}}$	$E_j^{\text{standby}}$
Washing machine	8:00	17:00	2 W	5 W
Tumble dryer	18:00	21:00	2 W	5 W
Dishwasher	10:00	18:00	2 W	5 W
CHP plant	0:00	n/A	2 W	5 W
Condensing boiler	0:00	n/A	2 W	5 W
Air-conditioning	0:00	n/A	2 W	5 W

Table F.4.: Technical specification of the air-conditioning. The air-conditioning possesses four different operation modes.

Cooling power	0.75 kW	1.5 kW	2.25 kW	3 kW
Electricity consumption	0.25 kW	0.5 kW	0.75 kW	1 kW
Minimum phase runtime	5 min			
Maximum phase runtime	333 min			
Minimum standby time	5 min			
Energy efficiency ratio	3			
Maximum number of phases	15			

Table F.5.: Technical specification of the CHP plant and the condensing boiler. The condensing boiler possesses four different operation modes.

	CHP plant	Condensing boiler			
Heating power	2 kW	2 kW	4 kW	8 kW	14 kW
Electricity consumption	-1 kW	15 kW	30 kW	60 kW	110 kW
Minimum phase runtime	5 min	5 min			
Maximum phase runtime	500 min	500 min			
Minimum standby time	10 min	10 min			
Energy conversion efficiency	70 % thermal, 25 % electrical	95 % thermal			
Maximum number of phases	15	15			

Table F.6.: Nomenclature of parameters, variables and objectives used for the BEMS optimization problem.

$\mathbf{a}^j$	Vector of operation modes for each phase
$a_{j,k,l}$	Operation mode $l$ of device $j$ in phase $k$
$A$	Area of the building ( $\text{m}^2$ )
$A^{\text{env}}$	Area of the building envelope ( $\text{m}^2$ )
$B(t)$	Carbon dioxide emissions in time slot $t$ (g)
$B$	Total carbon dioxide emissions the optimization horizon (g)
$c^{\text{air}}$	Specific heat capacity of air ( $\text{J kg}^{-1} \text{K}^{-1}$ )
$c^{\text{water}}$	Specific heat capacity of water ( $\text{J kg}^{-1} \text{K}^{-1}$ )
$C$	Total costs for energy consumption across the optimization horizon (ct)
$C(t)$	Energy costs that occur in time slot $t$ (ct)
$d_j$	Deadline of device $j$ , i.e. its latest time to finish
$D$	Average thermal discomfort across the optimization horizon ( $^{\circ}\text{C}$ )
$D(t)$	Thermal discomfort that occurs in time slot $t$ ( $^{\circ}\text{C}$ )
$e^b(t)$	$\text{CO}_2$ emissions of consuming a unit of electricity from the grid in time slot $t$ ( $\text{g kW}^{-1} \text{h}^{-1}$ )
$e^g$	$\text{CO}_2$ emissions of consuming a unit of natural gas ( $\text{g kW}^{-1} \text{h}^{-1}$ )
$h$	Height of the building (m)
$h^{\text{loss}}$	Heat transfer coefficient of the building ( $\text{W m}^{-2} \text{K}^{-1}$ )
$h^{\text{WS}_{\text{loss}}}$	Heat transfer coefficient of the hot water storage ( $\text{W m}^{-2} \text{K}^{-1}$ )
$J$	The set of all devices in the building
$j$	Identifier of a device
$k$	Identifier of a phase
$l$	Identifier of an operation mode
$n_j$	Number of operation cycles of device $j$
$E_j(t)$	Electricity consumed/generated by device $j$ in time slot $t$ (kW h)
$E_{j,k,l}(t^{\text{ref}})$	Electricity consumption/provision of device $j$ in phase $k$ in operation mode $l$ (kW h)
$E_j^{\text{standby}}$	Standby electricity consumption of device $j$ in a single time slot (kW h)
$G$	Total gas consumption across the optimization horizon (kW h)
$G_j(t)$	Natural gas consumption of device $j$ in time slot $t$ (kW h)
$G_{j,k,l}(t^{\text{ref}})$	Natural gas consumption of device $j$ in phase $k$ in operation mode $l$ (kW h)
$H$	The set of all time slots within the optimization horizon
$r_j$	Release time of device $j$ , i.e. its earliest time of execution
$\rho^{\text{air}}$	Density of air ( $\text{kg m}^{-3}$ )
$\rho^{\text{water}}$	Density of water ( $\text{kg m}^{-3}$ )
$\mathbf{p}^j$	Vector of lengths of each individual phase
$p_{j,k}$	Phase length of phase $k$ of device $j$
$p(t)$	Price paid per unit of electricity bought from/sold to the utility (ct $\text{kW}^{-1} \text{h}^{-1}$ )
$p^g(t)$	Price paid per unit of gas consumed (ct $\text{kW}^{-1} \text{h}^{-1}$ )
$p^b(t)$	Price paid per unit of electricity bought from to the utility (ct $\text{kW}^{-1} \text{h}^{-1}$ )
$p_{\text{CHP}}$	Price paid per unit of electricity sold generated by the micro-CHP plant (ct $\text{kW}^{-1} \text{h}^{-1}$ )
$p_{\text{PV}}$	Price paid per unit of electricity sold generated by the PV system (ct $\text{kW}^{-1} \text{h}^{-1}$ )
$Q_j(t)$	Heating/cooling generated by device $j$ in time slot $t$ (J)
$Q_{j,k,l}(t^{\text{ref}})$	Generated heat/cooling of device $j$ in phase $k$ in operation mode $l$ (J)
$Q^{\Delta}(t)$	Heating/cooling added to regulate temperature inside the building (J)
$Q^{\text{in}}(t)$	Heat generated by burning gas (J)
$Q^{\text{heat}}(t)$	Heat extracted from the hot water storage tank for heating (J)
$Q^{\text{loss}}(t)$	Heating/cooling loss caused by difference between inside and outside temperature (J)
$Q_{\text{WS}}^{\text{max}}$	Maximum heat extractable from the hot water storage (J)
$Q^{\text{NH}}(t)$	Required heat/cooling to maintain building temperature (J)
$Q^{\text{other}}(t)$	Waste heat generated by devices and residents (J)
$Q^{\text{water}}(t)$	Energy of warm water used by residents (J)
$\mathbf{s}^j$	Vector of delays for each individual phase
$s_{j,k}$	Delay of phase $k$ of device $j$
$t$	The variable identifying individual time slots
$t^{\text{ref}}$	Referred time in the load profile $E_{j,k,l}$
$T(t)$	Temperature in the building at time $t$

## F. BEMS Study Documentation

$T_{WS}^{\text{heat}}$	Minimum water temperature for warm water extraction
$T^{\text{l}}$	Lower thermal comfort threshold ( $^{\circ}\text{C}$ )
$T^{\text{max}}$	Maximum allowed temperature in building ( $^{\circ}\text{C}$ )
$T_{WS}^{\text{max}}$	Maximum allowed temperature of the hot water storage ( $^{\circ}\text{C}$ )
$T^{\text{min}}$	Minimum allowed temperature in building ( $^{\circ}\text{C}$ )
$T_{WS}^{\text{min}}$	Minimum allowed temperature of the hot water storage ( $^{\circ}\text{C}$ )
$T^{\text{out}}(t)$	Outside temperature at time $t$ ( $^{\circ}\text{C}$ )
$T^{\text{set}}$	Target temperature of the building set by the residents ( $^{\circ}\text{C}$ )
$T^{\text{u}}$	Upper thermal comfort threshold ( $^{\circ}\text{C}$ )
$T_{WS}$	Temperature of the hot water storage ( $^{\circ}\text{C}$ )
$V_{WS}$	Volume of the hot water storage ( $^{\circ}\text{C}$ )

---

# **G. Statistics**

## **G.1. Descriptive Statistics**

Table G.1.: Local search study – PD of CMA-ES. Median and IQR (as subscript) results for different step sizes  $\sigma$  across 100 runs. Best performances are colored in green, second-best performances in blue. Smallest PDs are achieved for step sizes of  $1e-3$  and  $1e-2$ . These values provide a balance between a narrow and broad search within the assigned basin.

	Sum of objectives					Nash				
	1e-5	1e-4	1e-3	1e-2	1e-1	1e-5	1e-4	1e-3	1e-2	1e-1
	DEB2DK $k = 1$	3.29e-3 <sub>1.61e-3</sub>	2.58e-1 <sub>2.45e-1</sub>	3.46e-4 <sub>4.06e-4</sub>	8.76e-4 <sub>1.21e-3</sub>	4.94e-1 <sub>2.26e-3</sub>	5.60e-3 <sub>5.98e-3</sub>	8.34e-4 <sub>4.77</sub>	1.18e-3 <sub>1.36e-3</sub>	2.50e-3 <sub>3.47e-3</sub>
DEB2DK $k = 2$	4.54e-3 <sub>8.61e-2</sub>	1.23e-1 <sub>1.23e-1</sub>	7.97e-4 <sub>5.65e-4</sub>	1.48e-3 <sub>1.19e-3</sub>	2.48e-1 <sub>1.97e-3</sub>	1.98e-1 <sub>1.97e-1</sub>	1.97e-1 <sub>1.97e-1</sub>	1.31e-3 <sub>1.04e-3</sub>	3.47e-3 <sub>2.66e-3</sub>	4.87e-3 <sub>3.73e-3</sub>
DEB2DK $k = 3$	3.73e-2 <sub>7.06e-2</sub>	7.27e-2 <sub>7.23e-2</sub>	9.18e-4 <sub>5.26e-4</sub>	2.35e-3 <sub>1.87e-3</sub>	1.45e-1 <sub>1.48e-3</sub>	1.08e-1 <sub>2.03e-1</sub>	1.05e-1 <sub>1.04e-1</sub>	1.37e-3 <sub>9.32e-4</sub>	3.42e-3 <sub>1.98e-3</sub>	2.07e-1 <sub>1.14e-3</sub>
DEB2DK $k = 4$	4.56e-2 <sub>4.60e-2</sub>	4.71e-2 <sub>4.68e-2</sub>	8.73e-4 <sub>4.92e-4</sub>	2.63e-3 <sub>4.63e-2</sub>	9.38e-2 <sub>1.46e-3</sub>	1.24e-1 <sub>8.18e-2</sub>	1.34e-1 <sub>5.53e-2</sub>	2.03e-3 <sub>1.63e-3</sub>	5.03e-3 <sub>4.47e-3</sub>	1.34e-1 <sub>8.41e-4</sub>
DEB3DK $k = 1$	4.65e-2 <sub>9.15e-2</sub>	4.53e-2 <sub>9.12e-2</sub>	2.70e-1 <sub>1.73e-1</sub>	2.52e-3 <sub>1.11e-1</sub>	1.12e-1 <sub>9.04e-4</sub>	2.81e-2 <sub>1.23e-2</sub>	2.92e-2 <sub>1.36e-2</sub>	1.98e-2 <sub>5.76</sub>	3.24e-3 <sub>2.68e-3</sub>	2.99e-3 <sub>2.70e-3</sub>
DEB3DK $k = 2$	5.78e-2 <sub>2.53e-2</sub>	6.03e-2 <sub>2.88e-2</sub>	9.76e-2 <sub>2.39e-2</sub>	1.90e-2 <sub>5.46e-2</sub>	2.20e-1 <sub>1.56e-2</sub>	6.63e-2 <sub>2.71e-2</sub>	7.11e-2 <sub>3.13e-2</sub>	3.30e-1 <sub>5.22</sub>	4.34e-2 <sub>2.16e-1</sub>	4.41e-2 <sub>1.70e-1</sub>
DO2DK $k = 1 s = 1$	2.58e-3 <sub>3.04e-3</sub>	1.23e-3 <sub>3.40e-3</sub>	7.05e-4 <sub>8.89e-4</sub>	8.63e-4 <sub>1.74e-3</sub>	1.19e-3 <sub>1.37e-3</sub>	2.44e-3 <sub>2.63e-3</sub>	2.50e-3 <sub>5.13</sub>	5.31e-4 <sub>6.49e-4</sub>	6.63e-4 <sub>1.08e-3</sub>	7.95e-4 <sub>8.72e-4</sub>
DO2DK $k = 2 s = 1$	5.31e-3 <sub>4.54e-3</sub>	1.50e-1 <sub>5.30e-2</sub>	1.87e-3 <sub>2.07e-3</sub>	4.08e-3 <sub>4.37e-3</sub>	1.50e-1 <sub>1.25e-3</sub>	4.07e-3 <sub>3.46e-3</sub>	1.54e-1 <sub>1.53e-1</sub>	7.90e-4 <sub>8.52e-4</sub>	2.05e-3 <sub>1.62e-3</sub>	2.38e-3 <sub>1.92e-3</sub>
DO2DK $k = 3 s = 1$	3.56e-3 <sub>2.22e-3</sub>	3.25e-3 <sub>1.39e-1</sub>	8.88e-4 <sub>6.69e-4</sub>	1.58e-3 <sub>1.37e-3</sub>	1.46e-1 <sub>1.47e-1</sub>	3.77e-3 <sub>5.41e-3</sub>	4.31e-3 <sub>4.60e-3</sub>	6.08e-4 <sub>6.96e-4</sub>	1.04e-3 <sub>1.10e-3</sub>	1.12e-3 <sub>1.56e-3</sub>
DO2DK $k = 4 s = 1$	5.97e-3 <sub>9.18e-2</sub>	1.39e-1 <sub>5.88e-2</sub>	1.69e-3 <sub>9.35e-2</sub>	3.36e-3 <sub>9.23e-2</sub>	1.57e-1 <sub>4.92e-4</sub>	4.36e-3 <sub>9.37e-2</sub>	9.55e-2 <sub>9.47e-2</sub>	6.03e-4 <sub>4.93e-4</sub>	1.01e-3 <sub>9.97e-4</sub>	1.28e-3 <sub>1.14e-3</sub>
DTLZ7	3.92e-2 <sub>1.15e-2</sub>	4.03e-2 <sub>1.28e-2</sub>	2.10e-1 <sub>1.03e-1</sub>	8.72e-3 <sub>3.48e-3</sub>	1.03e-2 <sub>5.10e-3</sub>	5.70 <sub>5.18</sub>	6.10 <sub>6.60</sub>	1.72e-1 <sub>1.53e-1</sub>	1.95e-2 <sub>7.72e-3</sub>	1.95e-2 <sub>2.96e-3</sub>
ZDT3	4.74e-3 <sub>1.17e-3</sub>	1.77e-1 <sub>1.13e-1</sub>	2.42e-4 <sub>1.99e-4</sub>	2.12e-4 <sub>1.40e-4</sub>	6.46e-2 <sub>6.01e-2</sub>	2.80 <sub>1.37e-1</sub>	2.82 <sub>2.75</sub>	4.10e-2 <sub>1.58e-3</sub>	4.10e-2 <sub>1.36e-3</sub>	2.16e-1 <sub>1.22e-1</sub>
	Angle utility					Tradeoff utility				
	1e-5	1e-4	1e-3	1e-2	1e-1	1e-5	1e-4	1e-3	1e-2	1e-1
DEB2DK $k = 1$	3.08e-3 <sub>5.42e-1</sub>	3.41e-4 <sub>1.10e-3</sub>	1.34e-4 <sub>1.73e-4</sub>	1.34e-4 <sub>1.73e-4</sub>	1.34e-4 <sub>1.73e-4</sub>	2.92e-3 <sub>1.37e-3</sub>	2.55e-3 <sub>1.26e-3</sub>	2.11e-3 <sub>3.49e-4</sub>	3.09e-3 <sub>1.42e-3</sub>	3.09e-3 <sub>1.42e-3</sub>
DEB2DK $k = 2$	2.09e-1 <sub>2.08e-1</sub>	2.08e-1 <sub>2.07e-1</sub>	1.53e-3 <sub>1.03e-3</sub>	3.38e-3 <sub>2.72e-3</sub>	4.52e-3 <sub>2.47e-3</sub>	3.48e-3 <sub>1.09e-3</sub>	2.47e-3 <sub>1.15e-3</sub>	1.62e-3 <sub>6.20e-4</sub>	3.72e-3 <sub>1.21e-3</sub>	3.72e-3 <sub>1.21e-3</sub>
DEB2DK $k = 3$	1.21e-1 <sub>2.27e-1</sub>	1.16e-1 <sub>1.15e-1</sub>	1.21e-3 <sub>1.65e-3</sub>	2.51e-3 <sub>1.65e-3</sub>	2.94e-3 <sub>1.67e-3</sub>	3.49e-3 <sub>1.05e-3</sub>	2.81e-3 <sub>1.05e-3</sub>	2.30e-3 <sub>8.60e-4</sub>	3.67e-3 <sub>9.60e-4</sub>	3.67e-3 <sub>9.60e-4</sub>
DEB2DK $k = 4$	1.49e-1 <sub>1.89e-1</sub>	7.91e-2 <sub>2.10e-1</sub>	1.37e-3 <sub>8.85e-4</sub>	2.07e-3 <sub>1.51e-3</sub>	4.06e-3 <sub>1.80e-3</sub>	3.63e-3 <sub>1.11e-3</sub>	2.62e-3 <sub>9.13e-4</sub>	2.21e-3 <sub>7.42e-4</sub>	3.80e-3 <sub>9.92e-4</sub>	3.80e-3 <sub>9.92e-4</sub>
DEB3DK $k = 1$	2.89e-2 <sub>1.86e-2</sub>	3.06e-2 <sub>1.89e-2</sub>	3.43e-2 <sub>2.84e-1</sub>	3.65e-3 <sub>2.68e-3</sub>	2.89e-3 <sub>1.20e-3</sub>	6.46e-2 <sub>1.62e-2</sub>	6.43e-2 <sub>1.66e-2</sub>	5.88e-2 <sub>1.25e-2</sub>	4.39e-2 <sub>1.54e-2</sub>	6.49e-2 <sub>1.57e-2</sub>
DEB3DK $k = 2$	6.50e-2 <sub>2.01e-2</sub>	6.47e-2 <sub>2.13e-2</sub>	3.13e-1 <sub>1.86e-1</sub>	5.22e-2 <sub>1.55e-1</sub>	1.94e-1 <sub>1.40e-1</sub>	1.19e-1 <sub>1.45e-2</sub>	1.19e-1 <sub>1.77e-2</sub>	1.20e-1 <sub>1.76e-2</sub>	1.23e-1 <sub>1.43e-2</sub>	1.19e-1 <sub>1.64e-2</sub>
DO2DK $k = 1 s = 1$	3.45e-3 <sub>4.45e-1</sub>	1.66e-2 <sub>1.35</sub>	1.55e-4 <sub>1.90e-4</sub>	1.60e-4 <sub>1.94e-4</sub>	1.50e-4 <sub>1.98e-4</sub>	2.63e-3 <sub>9.99e-4</sub>	2.23e-3 <sub>6.10e-4</sub>	3.97e-4 <sub>4.35e-3</sub>	3.11e-3 <sub>1.64e-3</sub>	3.11e-3 <sub>1.64e-3</sub>
DO2DK $k = 2 s = 1$	5.06e-3 <sub>5.23e-3</sub>	1.99 <sub>6.39</sub>	9.58e-5 <sub>8.59e-5</sub>	9.39e-5 <sub>8.45e-5</sub>	9.34e-5 <sub>8.46e-5</sub>	3.34e-3 <sub>1.44e-3</sub>	2.34e-3 <sub>1.22e-3</sub>	1.79e-3 <sub>1.27e-3</sub>	3.59e-3 <sub>1.60e-3</sub>	3.59e-3 <sub>1.60e-3</sub>
DO2DK $k = 3 s = 1$	8.62e-3 <sub>8.31</sub>	2.84 <sub>6.74</sub>	3.27e-4 <sub>1.46e-4</sub>	3.31e-4 <sub>1.55e-4</sub>	3.31e-4 <sub>1.48e-4</sub>	3.71e-3 <sub>1.34e-3</sub>	3.13e-3 <sub>1.51e-3</sub>	3.58e-3 <sub>1.27e-3</sub>	3.86e-3 <sub>1.28e-3</sub>	3.86e-3 <sub>1.28e-3</sub>
DO2DK $k = 4 s = 1$	3.81e-3 <sub>8.08</sub>	1.75 <sub>8.17e-1</sub>	7.12e-4 <sub>4.36e-4</sub>	7.23e-4 <sub>4.36e-4</sub>	7.23e-4 <sub>4.34e-4</sub>	3.98e-3 <sub>1.69e-3</sub>	2.97e-3 <sub>1.12e-3</sub>	1.88e-3 <sub>1.15e-3</sub>	4.04e-3 <sub>1.72e-3</sub>	4.04e-3 <sub>1.72e-3</sub>
DTLZ7	1.05 <sub>1.70</sub>	8.75e-1 <sub>6.84e-1</sub>	2.26e-1 <sub>6.38e-2</sub>	1.80e-1 <sub>7.36e-3</sub>	2.44e-1 <sub>1.52e-1</sub>	7.23e-2 <sub>2.16e-2</sub>	7.02e-2 <sub>2.03e-2</sub>	8.07e-2 <sub>2.90e-2</sub>	1.42e-1 <sub>3.42e-2</sub>	7.23e-2 <sub>2.16e-2</sub>
ZDT3	1.55 <sub>3.10e-1</sub>	7.41e-1 <sub>2.96e-2</sub>	1.91e-3 <sub>1.90e-4</sub>	1.95e-3 <sub>2.50e-4</sub>	3.68e-1 <sub>6.93e-3</sub>	4.30e-3 <sub>1.75e-3</sub>	2.74e-3 <sub>1.29e-3</sub>	1.27e-3 <sub>3.41e-4</sub>	4.66e-3 <sub>1.32e-3</sub>	4.66e-3 <sub>1.37e-3</sub>

Table G.2.: Local search study – PD of the GA. Median and IQR results for different distribution indices  $\eta$  across 100 runs. Best performances are colored in green, second-best performances in blue. Smallest PDs are achieved for a distribution index of 120, since estimates of the local scalarization optima found in the initial populations are already very close to the true optima. Further increasing  $\eta$  might improve the performance, however would narrow the search too much if the initial estimate would be located further away from the true optimum.

	Sum of objectives					Nash				
	20	30	40	80	120	20	30	40	80	120
	DEB2DK $k = 1$	4.93e-1 <sub>2.89e-3</sub>	4.91e-1 <sub>2.44e-1</sub>	2.48e-1 <sub>3.16e-1</sub>	2.18e-3 <sub>6.99e-4</sub>	2.08e-3 <sub>7.07e-4</sub>	2.62e-3 <sub>2.24e-3</sub>	2.17e-3 <sub>2.03e-3</sub>	2.19e-3 <sub>2.07e-3</sub>	2.11e-3 <sub>1.96e-3</sub>
DEB2DK $k = 2$	2.48e-1 <sub>3.35e-3</sub>	2.49e-1 <sub>2.23e-3</sub>	2.49e-1 <sub>2.08e-3</sub>	1.24e-1 <sub>1.23e-1</sub>	2.74e-3 <sub>31.90e-3</sub>	3.32e-3 <sub>31.57e-3</sub>	3.24e-3 <sub>31.97e-3</sub>	3.25e-3 <sub>31.95e-3</sub>	3.05e-3 <sub>31.74e-3</sub>	2.91e-3 <sub>31.67e-3</sub>
DEB2DK $k = 3$	1.45e-1 <sub>2.69e-3</sub>	1.45e-1 <sub>2.15e-3</sub>	1.45e-1 <sub>1.79e-3</sub>	1.44e-1 <sub>1.48e-3</sub>	7.37e-2 <sub>5.73e-2</sub>	2.05e-1 <sub>4.38e-3</sub>	1.05e-1 <sub>1.04e-1</sub>	4.12e-3 <sub>9.96e-2</sub>	3.23e-3 <sub>31.33e-3</sub>	3.14e-3 <sub>31.46e-3</sub>
DEB2DK $k = 4$	9.91e-2 <sub>1.04e-1</sub>	9.38e-2 <sub>2.20e-3</sub>	9.39e-2 <sub>1.74e-3</sub>	9.40e-2 <sub>1.33e-3</sub>	9.32e-2 <sub>2.25e-2</sub>	1.34e-1 <sub>1.85e-3</sub>	1.34e-1 <sub>1.50e-3</sub>	1.34e-1 <sub>1.27e-3</sub>	1.24e-1 <sub>1.656e-2</sub>	5.53e-3 <sub>6.48e-2</sub>
DEB3DK $k = 1$	1.25e-3 <sub>1.680e-3</sub>	1.19e-3 <sub>2.66e-3</sub>	1.16e-3 <sub>1.68e-2</sub>	3.19e-2 <sub>1.11e-1</sub>	8.09e-3 <sub>1.11e-1</sub>	7.90e-3 <sub>4.74e-3</sub>	5.47e-3 <sub>3.03e-3</sub>	4.59e-3 <sub>3.42e-3</sub>	3.75e-3 <sub>2.36e-3</sub>	3.75e-3 <sub>2.79e-3</sub>
DEB3DK $k = 2$	1.02e-1 <sub>7.45e-3</sub>	9.80e-2 <sub>3.65e-3</sub>	9.52e-2 <sub>1.36e-2</sub>	4.44e-2 <sub>3.74e-2</sub>	4.13e-2 <sub>3.80e-2</sub>	4.86e-2 <sub>3.71e-2</sub>	4.58e-2 <sub>3.87e-2</sub>	4.39e-2 <sub>3.75e-2</sub>	4.30e-2 <sub>3.76e-2</sub>	4.25e-2 <sub>3.87e-2</sub>
DO2DK $k = 1 s = 1$	1.79e-3 <sub>1.69e-3</sub>	1.70e-3 <sub>1.63e-3</sub>	1.41e-3 <sub>1.66e-3</sub>	1.32e-3 <sub>1.51e-3</sub>	1.19e-3 <sub>1.49e-3</sub>	1.96e-3 <sub>1.35e-3</sub>	1.53e-3 <sub>1.32e-3</sub>	1.27e-3 <sub>9.99e-4</sub>	1.16e-3 <sub>3.48e-4</sub>	1.31e-3 <sub>1.25e-3</sub>
DO2DK $k = 2 s = 1$	1.51e-2 <sub>1.14e-3</sub>	1.51e-1 <sub>1.50e-3</sub>	1.50e-1 <sub>1.96e-3</sub>	3.69e-3 <sub>3.87e-3</sub>	3.16e-3 <sub>2.41e-3</sub>	2.81e-3 <sub>1.67e-3</sub>	2.57e-3 <sub>1.50e-3</sub>	2.68e-3 <sub>1.54e-3</sub>	2.37e-3 <sub>1.47e-3</sub>	2.26e-3 <sub>1.50e-3</sub>
DO2DK $k = 3 s = 1$	1.46e-1 <sub>1.45e-1</sub>	3.11e-3 <sub>3.33e-3</sub>	2.60e-3 <sub>3.82e-3</sub>	2.46e-3 <sub>3.39e-3</sub>	2.19e-3 <sub>3.150e-3</sub>	2.00e-3 <sub>3.152e-3</sub>	2.02e-3 <sub>3.63e-3</sub>	2.01e-3 <sub>3.163e-3</sub>	1.92e-3 <sub>3.144e-3</sub>	1.56e-3 <sub>3.148e-3</sub>
DO2DK $k = 4 s = 1$	1.56e-1 <sub>1.67e-3</sub>	1.57e-1 <sub>1.80e-3</sub>	1.55e-1 <sub>1.619e-2</sub>	9.45e-2 <sub>1.81e-3</sub>	9.36e-2 <sub>9.22e-2</sub>	3.22e-3 <sub>9.28e-2</sub>	2.70e-3 <sub>3.30e-3</sub>	2.60e-3 <sub>3.12e-3</sub>	2.22e-3 <sub>3.121e-3</sub>	2.08e-3 <sub>3.83e-4</sub>
DTLZ7	2.26e-2 <sub>7.61e-3</sub>	1.87e-2 <sub>5.94e-3</sub>	1.63e-2 <sub>5.42e-3</sub>	1.06e-2 <sub>3.60e-3</sub>	8.78e-3 <sub>2.65e-3</sub>	3.08e-2 <sub>1.17e-1</sub>	2.23e-2 <sub>8.18e-3</sub>	2.16e-2 <sub>8.80e-3</sub>	2.04e-2 <sub>8.36e-3</sub>	1.98e-2 <sub>8.93e-3</sub>
ZDT3	7.06e-2 <sub>9.76e-2</sub>	6.45e-2 <sub>6.23e-2</sub>	3.41e-3 <sub>3.20e-2</sub>	2.46e-3 <sub>5.24e-4</sub>	1.61e-3 <sub>1.85e-4</sub>	2.16e-1 <sub>1.09e-1</sub>	1.07e-1 <sub>1.61e-3</sub>	4.56e-2 <sub>6.11e-2</sub>	4.28e-2 <sub>2.79e-3</sub>	4.21e-2 <sub>2.21e-3</sub>
	Angle utility					Tradeoff utility				
	20	30	40	80	120	20	30	40	80	120
	DEB2DK $k = 1$	1.96e-3 <sub>1.15e-3</sub>	1.83e-3 <sub>1.08e-3</sub>	1.70e-3 <sub>7.99e-4</sub>	1.48e-3 <sub>5.68e-4</sub>	1.24e-3 <sub>5.30e-4</sub>	2.56e-3 <sub>8.17e-4</sub>	2.49e-3 <sub>7.12e-4</sub>	2.51e-3 <sub>8.78e-4</sub>	2.29e-3 <sub>6.41e-4</sub>
DEB2DK $k = 2$	3.60e-3 <sub>31.36e-3</sub>	3.46e-3 <sub>31.54e-3</sub>	3.32e-3 <sub>31.75e-3</sub>	3.04e-3 <sub>31.68e-3</sub>	3.15e-3 <sub>31.89e-3</sub>	3.00e-3 <sub>31.19e-3</sub>	3.00e-3 <sub>39.49e-4</sub>	2.88e-3 <sub>31.10e-3</sub>	2.63e-3 <sub>38.40e-4</sub>	2.34e-3 <sub>36.76e-4</sub>
DEB2DK $k = 3$	3.83e-3 <sub>31.13e-1</sub>	3.10e-3 <sub>31.48e-3</sub>	3.08e-3 <sub>31.43e-3</sub>	2.91e-3 <sub>31.33e-3</sub>	2.49e-3 <sub>31.17e-3</sub>	3.83e-3 <sub>2.15e-2</sub>	3.32e-3 <sub>31.23e-3</sub>	3.02e-3 <sub>31.24e-3</sub>	2.72e-3 <sub>31.44e-4</sub>	2.40e-3 <sub>36.40e-4</sub>
DEB2DK $k = 4$	8.12e-2 <sub>7.80e-2</sub>	3.63e-3 <sub>3.50e-3</sub>	3.29e-3 <sub>3.19e-3</sub>	3.11e-3 <sub>3.730e-4</sub>	2.97e-3 <sub>9.76e-4</sub>	1.39e-2 <sub>2.60e-2</sub>	4.15e-3 <sub>3.55e-2</sub>	4.07e-3 <sub>31.40e-2</sub>	2.79e-3 <sub>36.86e-4</sub>	2.57e-3 <sub>35.65e-4</sub>
DEB3DK $k = 1$	5.06e-2 <sub>2.76e-1</sub>	7.79e-3 <sub>3.52e-3</sub>	6.41e-3 <sub>3.87e-3</sub>	4.92e-3 <sub>2.65e-3</sub>	3.73e-3 <sub>3.27e-3</sub>	4.54e-2 <sub>1.33e-2</sub>	4.12e-2 <sub>2.69e-2</sub>	4.19e-2 <sub>21.43e-2</sub>	4.46e-2 <sub>21.23e-2</sub>	4.53e-2 <sub>21.33e-2</sub>
DEB3DK $k = 2$	4.31e-2 <sub>2.46e-2</sub>	4.04e-2 <sub>2.63e-2</sub>	4.05e-2 <sub>2.55e-2</sub>	3.96e-2 <sub>2.32e-2</sub>	3.96e-2 <sub>2.55e-2</sub>	1.57e-1 <sub>2.39e-2</sub>	1.46e-1 <sub>2.32e-2</sub>	1.36e-1 <sub>1.62e-2</sub>	1.20e-1 <sub>1.90e-2</sub>	1.16e-1 <sub>1.69e-2</sub>
DO2DK $k = 1 s = 1$	1.37e-3 <sub>31.56e-3</sub>	1.29e-3 <sub>31.22e-3</sub>	9.91e-4 <sub>4.740e-4</sub>	1.03e-3 <sub>3.677e-4</sub>	9.34e-4 <sub>4.655e-4</sub>	2.37e-3 <sub>8.96e-4</sub>	2.37e-3 <sub>9.32e-4</sub>	2.26e-3 <sub>7.06e-4</sub>	2.23e-3 <sub>8.95e-4</sub>	2.13e-3 <sub>7.58e-4</sub>
DO2DK $k = 2 s = 1$	2.81e-3 <sub>2.42e-3</sub>	2.32e-3 <sub>31.44e-3</sub>	2.14e-3 <sub>31.50e-3</sub>	1.51e-3 <sub>3.778e-4</sub>	1.29e-3 <sub>31.50e-4</sub>	2.68e-3 <sub>31.23e-3</sub>	2.51e-3 <sub>8.08e-4</sub>	2.40e-3 <sub>8.90e-4</sub>	2.15e-3 <sub>9.49e-4</sub>	1.91e-3 <sub>7.49e-4</sub>
DO2DK $k = 3 s = 1$	4.19e-3 <sub>3.35e-3</sub>	3.80e-3 <sub>2.97e-3</sub>	3.14e-3 <sub>2.25e-3</sub>	2.30e-3 <sub>9.88e-4</sub>	1.66e-3 <sub>9.69e-4</sub>	4.11e-3 <sub>31.42e-2</sub>	3.07e-3 <sub>9.62e-3</sub>	3.05e-3 <sub>7.21e-3</sub>	2.57e-3 <sub>31.20e-3</sub>	2.30e-3 <sub>31.06e-3</sub>
DO2DK $k = 4 s = 1$	2.18e-3 <sub>2.09e-3</sub>	1.96e-3 <sub>31.96e-3</sub>	1.87e-3 <sub>2.21e-3</sub>	1.45e-3 <sub>31.51e-3</sub>	1.29e-3 <sub>31.02e-3</sub>	4.03e-3 <sub>5.93e-2</sub>	3.42e-3 <sub>2.87e-3</sub>	3.13e-3 <sub>31.45e-3</sub>	2.46e-3 <sub>31.01e-3</sub>	2.26e-3 <sub>38.39e-4</sub>
DTLZ7	1.82e-1 <sub>1.658e-3</sub>	1.82e-1 <sub>1.652e-3</sub>	1.83e-1 <sub>1.669e-3</sub>	1.85e-1 <sub>1.825e-3</sub>	1.79e-1 <sub>1.35e-2</sub>	1.51e-1 <sub>2.61e-2</sub>	1.68e-1 <sub>1.29e-2</sub>	1.71e-1 <sub>8.25e-3</sub>	1.72e-1 <sub>4.11e-3</sub>	1.64e-1 <sub>2.54e-2</sub>
ZDT3	1.26e-1 <sub>1.600e-2</sub>	5.13e-3 <sub>6.24e-2</sub>	4.38e-3 <sub>31.29e-3</sub>	3.39e-3 <sub>3.741e-4</sub>	2.72e-3 <sub>6.90e-4</sub>	4.37e-3 <sub>2.15e-2</sub>	2.33e-2 <sub>2.32e-2</sub>	2.53e-2 <sub>2.33e-2</sub>	2.54e-2 <sub>9.43e-4</sub>	2.49e-2 <sub>2.38e-2</sub>

Table G.3.: Local search study – PD of the HC. Median and IQR (as subscript) results for different distribution indices  $\eta$  across 100 runs. Best performances are colored in green, second-best performances in blue. Smallest PDs are achieved for a distribution index of 120, since estimates of the local scalarization optima found in the initial populations are already very close to the true optima. Further increasing  $\eta$  might improve the performance, however would narrow the search too much if the initial estimate would be located further away from the true optimum. Such a performance deterioration is already partially observable for tradeoff utility.

	Sum of objectives					Nash				
	20	30	40	80	120	20	30	40	80	120
	DEB2DK $k = 1$	4.97e-1 <sub>2.80e-4</sub>	4.97e-1 <sub>6.67e-4</sub>	2.49e-1 <sub>2.48e-1</sub>	2.08e-5 <sub>3.24e-5</sub>	1.26e-5 <sub>2.20e-5</sub>	2.18e-4 <sub>3.58e-4</sub>	2.05e-4 <sub>2.35e-4</sub>	1.24e-4 <sub>2.35e-4</sub>	5.69e-5 <sub>1.04e-4</sub>
DEB2DK $k = 2$	2.49e-1 <sub>2.63e-4</sub>	2.49e-1 <sub>1.24e-4</sub>	2.49e-1 <sub>1.22e-4</sub>	2.33e-1 <sub>1.25e-1</sub>	7.04e-5 <sub>1.25e-1</sub>	4.04e-4 <sub>4.46e-4</sub>	2.75e-4 <sub>2.90e-4</sub>	2.28e-4 <sub>1.71e-4</sub>	1.40e-4 <sub>1.10e-4</sub>	1.20e-4 <sub>8.06e-5</sub>
DEB2DK $k = 3$	1.45e-1 <sub>3.18e-4</sub>	1.45e-1 <sub>1.89e-4</sub>	1.45e-1 <sub>1.33e-4</sub>	1.45e-1 <sub>6.65e-5</sub>	1.24e-1 <sub>7.24e-2</sub>	2.08e-1 <sub>3.27e-4</sub>	1.04e-1 <sub>1.04e-1</sub>	9.68e-2 <sub>1.04e-1</sub>	1.81e-4 <sub>1.21e-4</sub>	1.67e-4 <sub>9.16e-5</sub>
DEB2DK $k = 4$	1.99e-1 <sub>1.05e-1</sub>	9.35e-2 <sub>1.83e-4</sub>	9.35e-2 <sub>1.08e-4</sub>	9.35e-2 <sub>7.05e-5</sub>	9.35e-2 <sub>8.72e-5</sub>	1.34e-1 <sub>1.78e-4</sub>	1.34e-1 <sub>1.17e-4</sub>	1.34e-1 <sub>1.09e-4</sub>	1.34e-1 <sub>1.25e-4</sub>	3.73e-2 <sub>6.70e-2</sub>
DEB3DK $k = 1$	1.13e-1 <sub>1.74e-4</sub>	1.13e-1 <sub>8.76e-5</sub>	1.13e-1 <sub>9.46e-5</sub>	1.44e-4 <sub>1.13e-1</sub>	1.14e-4 <sub>1.13e-1</sub>	2.98e-3 <sub>3.33e-4</sub>	3.10e-3 <sub>6.99e-4</sub>	3.16e-3 <sub>5.79e-4</sub>	3.15e-3 <sub>3.33e-4</sub>	3.15e-3 <sub>6.45e-4</sub>
DEB3DK $k = 2$	9.09e-2 <sub>2.08e-4</sub>	9.09e-2 <sub>1.02e-2</sub>	9.09e-2 <sub>1.82e-4</sub>	3.98e-2 <sub>3.77e-2</sub>	1.57e-2 <sub>2.41e-2</sub>	4.24e-2 <sub>3.82e-2</sub>	4.23e-2 <sub>3.83e-2</sub>	4.25e-2 <sub>3.83e-2</sub>	2.40e-2 <sub>3.83e-2</sub>	4.26e-2 <sub>3.83e-2</sub>
DO2DK $k = 1 s = 1$	1.06e-4 <sub>2.20e-4</sub>	7.92e-5 <sub>1.30e-4</sub>	8.27e-5 <sub>1.09e-4</sub>	4.35e-5 <sub>7.28e-5</sub>	2.74e-5 <sub>6.04e-5</sub>	1.00e-4 <sub>1.38e-4</sub>	8.93e-5 <sub>1.35e-4</sub>	5.84e-5 <sub>1.03e-4</sub>	3.18e-5 <sub>6.14e-5</sub>	2.70e-5 <sub>5.82e-5</sub>
DO2DK $k = 2 s = 1$	1.50e-1 <sub>1.46e-4</sub>	1.50e-1 <sub>1.17e-4</sub>	1.50e-1 <sub>7.52e-5</sub>	1.12e-4 <sub>1.84e-4</sub>	6.78e-5 <sub>7.80e-5</sub>	2.71e-4 <sub>2.92e-4</sub>	1.93e-4 <sub>2.22e-4</sub>	1.47e-4 <sub>1.54e-4</sub>	9.14e-5 <sub>8.00e-5</sub>	6.80e-5 <sub>6.85e-5</sub>
DO2DK $k = 3 s = 1$	1.48e-1 <sub>1.65e-3</sub>	2.72e-4 <sub>3.77e-4</sub>	1.55e-4 <sub>2.12e-4</sub>	8.60e-5 <sub>9.20e-5</sub>	6.88e-5 <sub>7.04e-5</sub>	2.94e-4 <sub>4.49e-4</sub>	1.94e-4 <sub>2.27e-4</sub>	1.55e-4 <sub>2.46e-4</sub>	7.12e-5 <sub>1.27e-4</sub>	4.99e-5 <sub>8.80e-5</sub>
DO2DK $k = 4 s = 1$	1.58e-1 <sub>2.24e-4</sub>	1.58e-1 <sub>2.42e-4</sub>	1.58e-1 <sub>4.27e-4</sub>	9.40e-2 <sub>8.14e-5</sub>	9.39e-2 <sub>9.39e-2</sub>	4.34e-2 <sub>9.52e-2</sub>	2.64e-2 <sub>4.35e-4</sub>	2.05e-2 <sub>4.82e-4</sub>	9.57e-5 <sub>1.10e-4</sub>	6.46e-5 <sub>5.53e-5</sub>
DTLZ7	7.02e-3 <sub>4.18e-4</sub>	7.07e-3 <sub>2.69e-4</sub>	7.06e-3 <sub>2.02e-4</sub>	7.09e-3 <sub>1.23e-4</sub>	7.12e-3 <sub>7.31e-5</sub>	1.54e-1 <sub>1.07e-1</sub>	2.30e-2 <sub>8.87e-3</sub>	2.32e-2 <sub>8.50e-3</sub>	2.34e-2 <sub>1.15e-3</sub>	2.36e-2 <sub>8.57e-3</sub>
ZDT3	1.86e-1 <sub>6.55e-2</sub>	6.45e-2 <sub>5.25e-3</sub>	3.77e-4 <sub>6.42e-2</sub>	1.51e-4 <sub>1.03e-4</sub>	9.69e-5 <sub>5.79e-5</sub>	2.16e-1 <sub>1.28e-3</sub>	1.06e-1 <sub>4.48e-4</sub>	4.38e-2 <sub>6.28e-2</sub>	4.10e-2 <sub>1.83e-3</sub>	4.08e-2 <sub>1.47e-3</sub>
	Angle utility					Tradeoff utility				
	20	30	40	80	120	20	30	40	80	120
	DEB2DK $k = 1$	2.69e-4 <sub>4.01e-4</sub>	2.22e-4 <sub>3.01e-4</sub>	1.81e-4 <sub>2.28e-4</sub>	1.52e-4 <sub>2.32e-4</sub>	1.36e-4 <sub>1.70e-4</sub>	1.41e-4 <sub>2.39e-4</sub>	1.00e-4 <sub>2.09e-4</sub>	1.21e-4 <sub>1.66e-4</sub>	1.35e-4 <sub>1.91e-4</sub>
DEB2DK $k = 2$	3.76e-4 <sub>4.17e-4</sub>	3.32e-4 <sub>2.18e-4</sub>	2.77e-4 <sub>3.03e-4</sub>	1.97e-4 <sub>1.34e-4</sub>	1.81e-4 <sub>3.75e-5</sub>	4.87e-5 <sub>9.96e-2</sub>	4.27e-2 <sub>2.83e-2</sub>	2.50e-4 <sub>4.17e-4</sub>	2.38e-4 <sub>1.16e-4</sub>	2.40e-4 <sub>3.92e-5</sub>
DEB2DK $k = 3$	1.15e-1 <sub>1.15e-1</sub>	4.58e-4 <sub>3.48e-4</sub>	4.12e-4 <sub>2.36e-4</sub>	3.00e-4 <sub>2.19e-4</sub>	2.82e-4 <sub>1.98e-4</sub>	1.45e-1 <sub>1.22e-3</sub>	9.56e-2 <sub>4.92e-2</sub>	4.72e-2 <sub>1.63e-4</sub>	2.41e-2 <sub>2.31e-2</sub>	1.19e-3 <sub>3.53e-4</sub>
DEB2DK $k = 4$	1.57e-1 <sub>7.85e-2</sub>	8.62e-4 <sub>7.86e-2</sub>	3.31e-4 <sub>2.43e-4</sub>	1.87e-4 <sub>4.07e-4</sub>	1.59e-4 <sub>2.29e-5</sub>	1.04e-1 <sub>1.39e-3</sub>	1.04e-1 <sub>1.28e-3</sub>	7.49e-2 <sub>2.28e-2</sub>	3.17e-2 <sub>1.84e-4</sub>	1.64e-2 <sub>2.45e-2</sub>
DEB3DK $k = 1$	2.90e-1 <sub>9.13e-2</sub>	1.36e-3 <sub>3.62e-4</sub>	1.45e-3 <sub>6.13e-4</sub>	1.42e-3 <sub>4.70e-4</sub>	1.46e-3 <sub>5.19e-4</sub>	4.40e-2 <sub>1.63e-2</sub>	4.27e-2 <sub>2.13e-2</sub>	4.28e-2 <sub>2.14e-2</sub>	4.39e-2 <sub>2.13e-2</sub>	4.56e-2 <sub>2.13e-2</sub>
DEB3DK $k = 2$	4.06e-2 <sub>2.70e-2</sub>	4.06e-2 <sub>2.73e-2</sub>	4.07e-2 <sub>2.71e-2</sub>	4.10e-2 <sub>2.71e-2</sub>	4.10e-2 <sub>2.72e-2</sub>	2.06e-1 <sub>2.79e-2</sub>	1.77e-1 <sub>3.18e-2</sub>	1.49e-1 <sub>1.95e-2</sub>	1.20e-1 <sub>1.98e-2</sub>	1.16e-1 <sub>1.88e-2</sub>
DO2DK $k = 1 s = 1$	2.92e-4 <sub>4.26e-4</sub>	2.32e-4 <sub>4.10e-4</sub>	1.61e-4 <sub>3.34e-4</sub>	1.41e-4 <sub>2.41e-4</sub>	1.12e-4 <sub>2.11e-4</sub>	6.36e-5 <sub>1.77e-4</sub>	8.84e-5 <sub>2.09e-4</sub>	9.00e-5 <sub>2.33e-4</sub>	1.35e-4 <sub>3.14e-4</sub>	1.25e-4 <sub>3.02e-4</sub>
DO2DK $k = 2 s = 1$	2.75e-4 <sub>4.68e-4</sub>	1.63e-4 <sub>2.58e-4</sub>	1.04e-4 <sub>1.77e-4</sub>	7.87e-5 <sub>1.12e-4</sub>	7.18e-5 <sub>8.23e-5</sub>	2.30e-1 <sub>2.30e-1</sub>	4.48e-4 <sub>4.12e-4</sub>	4.54e-4 <sub>3.81e-4</sub>	5.12e-4 <sub>3.34e-4</sub>	5.20e-4 <sub>3.91e-4</sub>
DO2DK $k = 3 s = 1$	3.65e-4 <sub>3.80e-4</sub>	2.71e-4 <sub>2.95e-4</sub>	2.46e-4 <sub>2.12e-4</sub>	2.01e-4 <sub>1.93e-4</sub>	2.15e-4 <sub>2.25e-4</sub>	1.18e-1 <sub>4.95e-4</sub>	1.18e-1 <sub>1.01e-3</sub>	1.53e-2 <sub>2.103e-1</sub>	1.52e-2 <sub>3.53e-4</sub>	1.52e-2 <sub>3.45e-4</sub>
DO2DK $k = 4 s = 1$	1.56e-3 <sub>1.57e-3</sub>	1.19e-3 <sub>1.13e-3</sub>	1.19e-3 <sub>3.71e-4</sub>	9.34e-4 <sub>6.78e-4</sub>	8.71e-4 <sub>5.85e-4</sub>	8.19e-2 <sub>6.09e-2</sub>	8.15e-2 <sub>3.41e-4</sub>	8.15e-2 <sub>3.75e-4</sub>	6.00e-4 <sub>2.77e-4</sub>	5.72e-4 <sub>2.20e-4</sub>
DTLZ7	1.95e-1 <sub>1.63e-2</sub>	1.96e-1 <sub>1.66e-2</sub>	1.95e-1 <sub>1.43e-2</sub>	1.95e-1 <sub>1.35e-2</sub>	1.95e-1 <sub>1.22e-2</sub>	2.11e-1 <sub>1.13e-3</sub>	1.94e-1 <sub>1.80e-2</sub>	1.75e-1 <sub>1.75e-2</sub>	1.75e-1 <sub>1.26e-3</sub>	1.63e-1 <sub>1.34e-2</sub>
ZDT3	1.27e-1 <sub>1.21e-1</sub>	6.19e-2 <sub>6.60e-2</sub>	1.88e-3 <sub>3.21e-4</sub>	1.85e-3 <sub>3.62e-4</sub>	1.84e-3 <sub>3.04e-4</sub>	2.48e-2 <sub>2.19e-4</sub>	2.47e-2 <sub>1.61e-4</sub>	2.47e-2 <sub>2.12e-4</sub>	2.48e-2 <sub>3.38e-5</sub>	2.48e-2 <sub>6.98e-5</sub>

Table G.4.: Local search study – PD of the PSA. Median and IQR (as subscript) results for different inertia weights  $w$  across 100 runs. Best performances are colored in green, second-best performances in blue. Smallest PDs are achieved for an inertia weight of 0.75, which gives a large weight to the personal and global best position during the search.

	Sum of objectives					Nash				
	0.1	0.25	0.5	0.75	1	0.1	0.25	0.5	0.75	1
	DEB2DK $k = 1$	2.35e-3 <sub>3.70e-4</sub>	2.18e-3 <sub>3.80e-4</sub>	1.30e-3 <sub>4.20e-4</sub>	1.80e-9 <sub>1.09e-9</sub>	3.40e-3 <sub>4.92e-3</sub>	1.63e-3 <sub>3.85e-4</sub>	1.52e-3 <sub>3.83e-4</sub>	1.25e-3 <sub>6.45e-4</sub>	1.22e-5 <sub>1.57e-5</sub>
DEB2DK $k = 2$	2.33e-3 <sub>3.73e-4</sub>	2.18e-3 <sub>4.00e-4</sub>	1.49e-3 <sub>4.53e-4</sub>	3.67e-9 <sub>5.58e-10</sub>	5.64e-3 <sub>1.35e-2</sub>	2.02e-3 <sub>3.11e-4</sub>	1.95e-3 <sub>4.05e-4</sub>	1.63e-3 <sub>5.62e-4</sub>	9.88e-5 <sub>1.50e-5</sub>	1.91e-1 <sub>1.83e-1</sub>
DEB2DK $k = 3$	2.17e-3 <sub>4.10e-4</sub>	2.09e-3 <sub>3.49e-4</sub>	1.58e-3 <sub>4.72e-4</sub>	4.04e-9 <sub>1.11e-10</sub>	1.34e-1 <sub>6.70e-3</sub>	1.99e-3 <sub>2.98e-4</sub>	1.94e-3 <sub>3.26e-4</sub>	1.74e-3 <sub>6.03e-4</sub>	1.30e-4 <sub>2.38e-5</sub>	1.98e-1 <sub>1.22e-2</sub>
DEB2DK $k = 4$	2.19e-3 <sub>3.54e-4</sub>	2.09e-3 <sub>3.86e-4</sub>	1.62e-3 <sub>3.79e-4</sub>	5.20e-9 <sub>3.63e-10</sub>	9.54e-2 <sub>2.67e-3</sub>	2.24e-3 <sub>5.32e-4</sub>	2.10e-3 <sub>5.06e-4</sub>	6.78e-2 <sub>6.63e-2</sub>	1.34e-1 <sub>1.79e-7</sub>	1.36e-1 <sub>9.13e-3</sub>
DEB3DK $k = 1$	4.03e-2 <sub>1.01e-1</sub>	3.68e-2 <sub>9.91e-2</sub>	2.04e-2 <sub>1.03e-1</sub>	1.73e-4 <sub>1.13e-1</sub>	1.32e-2 <sub>1.05e-1</sub>	1.91e-2 <sub>3.34e-3</sub>	1.82e-2 <sub>3.09e-3</sub>	1.32e-2 <sub>4.86e-3</sub>	3.27e-3 <sub>7.02e-4</sub>	2.02e-2 <sub>2.85e-2</sub>
DEB3DK $k = 2$	4.86e-2 <sub>3.52e-2</sub>	4.68e-2 <sub>3.41e-2</sub>	3.54e-2 <sub>2.86e-2</sub>	1.60e-2 <sub>3.75e-2</sub>	2.79e-1 <sub>1.31e-5</sub>	5.80e-2 <sub>3.16e-2</sub>	5.70e-2 <sub>3.23e-2</sub>	5.29e-2 <sub>3.48e-2</sub>	4.26e-2 <sub>3.78e-2</sub>	2.27e-1 <sub>1.73e-2</sub>
DO2DK $k = 1 s = 1$	1.09e-3 <sub>4.15e-4</sub>	1.07e-3 <sub>4.25e-4</sub>	9.73e-4 <sub>8.58e-4</sub>	5.31e-9 <sub>1.17e-9</sub>	5.06e-3 <sub>3.01e-2</sub>	1.09e-3 <sub>2.64e-4</sub>	1.06e-3 <sub>2.78e-4</sub>	9.51e-4 <sub>4.5.32e-4</sub>	1.66e-5 <sub>2.19e-5</sub>	5.14e-3 <sub>8.67e-3</sub>
DO2DK $k = 2 s = 1$	1.90e-3 <sub>4.83e-4</sub>	1.85e-3 <sub>6.39e-4</sub>	2.14e-3 <sub>3.2.17e-3</sub>	1.50e-1 <sub>9.60e-10</sub>	1.50e-1 <sub>2.66e-3</sub>	1.77e-3 <sub>3.11e-4</sub>	1.70e-3 <sub>3.2.79e-4</sub>	1.47e-3 <sub>4.1.76e-4</sub>	4.43e-5 <sub>1.02e-5</sub>	1.53e-1 <sub>5.20e-3</sub>
DO2DK $k = 3 s = 1$	1.95e-3 <sub>3.90e-4</sub>	1.96e-3 <sub>5.01e-4</sub>	1.51e-3 <sub>6.18e-4</sub>	5.67e-9 <sub>1.09e-9</sub>	1.45e-1 <sub>1.25e-2</sub>	1.47e-3 <sub>3.14e-4</sub>	1.42e-3 <sub>3.91e-4</sub>	1.14e-3 <sub>4.49e-4</sub>	2.98e-5 <sub>1.96e-5</sub>	9.83e-3 <sub>1.76e-2</sub>
DO2DK $k = 4 s = 1$	2.17e-3 <sub>9.27e-2</sub>	2.33e-3 <sub>9.25e-2</sub>	9.42e-2 <sub>9.21e-4</sub>	9.39e-2 <sub>7.78e-10</sub>	2.50e-1 <sub>9.78e-3</sub>	1.65e-3 <sub>3.41e-4</sub>	1.59e-3 <sub>3.20e-4</sub>	1.28e-3 <sub>3.64e-4</sub>	2.05e-5 <sub>1.55e-5</sub>	8.93e-2 <sub>8.23e-2</sub>
DTLZ7	2.53e-2 <sub>5.11e-3</sub>	2.38e-2 <sub>4.28e-3</sub>	1.86e-2 <sub>4.89e-3</sub>	7.12e-3 <sub>1.17e-9</sub>	2.17e-2 <sub>1.05e-2</sub>	2.92e-2 <sub>9.08e-3</sub>	2.73e-2 <sub>7.61e-3</sub>	2.33e-2 <sub>7.65e-3</sub>	2.22e-2 <sub>2.28e-3</sub>	4.58e-1 <sub>1.37e-2</sub>
ZDT3	3.97e-3 <sub>9.36e-4</sub>	3.91e-3 <sub>8.32e-4</sub>	2.96e-3 <sub>7.48e-4</sub>	6.27e-9 <sub>1.89e-10</sub>	5.74e-1 <sub>5.07e-2</sub>	6.49e-3 <sub>1.39e-3</sub>	1.76e-2 <sub>4.16e-3</sub>	4.28e-2 <sub>2.18e-3</sub>	4.08e-1 <sub>2.32e-3</sub>	2.15e-1 <sub>1.63e-2</sub>
	Angle utility					Tradeoff utility				
	0.1	0.25	0.5	0.75	1	0.1	0.25	0.5	0.75	1
	DEB2DK $k = 1$	1.73e-3 <sub>4.07e-4</sub>	1.71e-3 <sub>3.88e-4</sub>	1.67e-3 <sub>4.20e-4</sub>	2.78e-4 <sub>6.63e-4</sub>	9.13e-3 <sub>3.1.77e-2</sub>	2.71e-3 <sub>5.11e-4</sub>	2.47e-3 <sub>4.70e-4</sub>	1.50e-3 <sub>3.96e-4</sub>	1.75e-4 <sub>2.00e-4</sub>
DEB2DK $k = 2$	2.08e-3 <sub>3.34e-4</sub>	2.03e-3 <sub>3.71e-4</sub>	1.68e-3 <sub>3.7.15e-4</sub>	1.88e-4 <sub>2.89e-5</sub>	2.06e-1 <sub>1.46e-2</sub>	2.36e-3 <sub>3.21e-4</sub>	2.26e-3 <sub>3.22e-4</sub>	1.69e-3 <sub>4.45e-4</sub>	3.01e-4 <sub>4.39e-4</sub>	1.29e-1 <sub>4.859e-2</sub>
DEB2DK $k = 3$	2.07e-3 <sub>3.2.86e-4</sub>	2.08e-3 <sub>3.4.40e-4</sub>	1.87e-3 <sub>3.5.90e-4</sub>	2.94e-4 <sub>2.16e-4</sub>	2.18e-1 <sub>1.62e-3</sub>	2.41e-3 <sub>3.96e-4</sub>	2.33e-3 <sub>3.4.45e-4</sub>	1.89e-3 <sub>3.4.43e-4</sub>	1.04e-3 <sub>3.24e-4</sub>	1.02e-1 <sub>5.69e-2</sub>
DEB2DK $k = 4$	2.08e-3 <sub>3.330e-4</sub>	2.02e-3 <sub>3.90e-4</sub>	1.71e-3 <sub>3.5.10e-4</sub>	1.42e-4 <sub>2.54e-5</sub>	1.57e-1 <sub>6.95e-3</sub>	2.36e-3 <sub>3.02e-4</sub>	2.24e-3 <sub>3.56e-4</sub>	1.83e-3 <sub>3.64e-4</sub>	1.11e-3 <sub>3.52e-4</sub>	8.26e-2 <sub>3.67e-2</sub>
DEB3DK $k = 1$	2.37e-2 <sub>4.55e-3</sub>	2.21e-2 <sub>4.45e-3</sub>	1.52e-2 <sub>4.14e-3</sub>	1.50e-3 <sub>6.41e-4</sub>	1.83e-2 <sub>1.68e-2</sub>	6.05e-2 <sub>2.1.22e-2</sub>	5.99e-2 <sub>1.17e-2</sub>	5.36e-2 <sub>1.45e-2</sub>	4.40e-2 <sub>2.57e-2</sub>	4.52e-2 <sub>1.60e-2</sub>
DEB3DK $k = 2$	5.22e-2 <sub>5.19e-3</sub>	5.17e-2 <sub>4.43e-3</sub>	4.77e-2 <sub>6.60e-3</sub>	4.12e-2 <sub>3.30e-3</sub>	2.37e-1 <sub>3.36e-2</sub>	1.19e-1 <sub>1.50e-2</sub>	1.19e-1 <sub>1.63e-2</sub>	1.18e-1 <sub>1.79e-2</sub>	1.21e-1 <sub>1.98e-2</sub>	1.66e-1 <sub>2.33e-2</sub>
DO2DK $k = 1 s = 1$	2.10e-3 <sub>5.94e-4</sub>	2.11e-3 <sub>6.07e-4</sub>	1.96e-3 <sub>5.87e-4</sub>	2.21e-4 <sub>3.98e-4</sub>	1.03e-2 <sub>1.38e-2</sub>	2.72e-3 <sub>7.96e-4</sub>	2.61e-3 <sub>7.89e-4</sub>	1.58e-3 <sub>5.92e-4</sub>	2.00e-4 <sub>3.00e-4</sub>	4.79e-4 <sub>7.22e-4</sub>
DO2DK $k = 2 s = 1$	2.06e-3 <sub>4.66e-4</sub>	2.06e-3 <sub>4.68e-4</sub>	1.98e-3 <sub>5.11e-4</sub>	1.80e-4 <sub>8.11e-4</sub>	7.82e-3 <sub>1.15e-2</sub>	2.44e-3 <sub>7.36e-4</sub>	2.35e-3 <sub>6.84e-4</sub>	1.69e-3 <sub>5.40e-4</sub>	5.84e-4 <sub>3.17e-4</sub>	1.12e-1 <sub>5.93e-3</sub>
DO2DK $k = 3 s = 1$	2.84e-3 <sub>6.14e-4</sub>	2.82e-3 <sub>6.54e-4</sub>	2.66e-3 <sub>6.47e-4</sub>	3.74e-4 <sub>4.17e-4</sub>	1.17e-2 <sub>2.1.57e-2</sub>	2.58e-3 <sub>6.45e-4</sub>	2.50e-3 <sub>5.82e-4</sub>	2.01e-3 <sub>8.02e-4</sub>	1.45e-2 <sub>1.29e-2</sub>	8.30e-2 <sub>6.52e-3</sub>
DO2DK $k = 4 s = 1$	2.57e-3 <sub>7.54e-4</sub>	2.56e-3 <sub>7.30e-4</sub>	2.23e-3 <sub>8.47e-4</sub>	6.62e-4 <sub>4.44e-4</sub>	1.67e-2 <sub>3.81e-2</sub>	2.60e-3 <sub>4.73e-4</sub>	2.43e-3 <sub>5.28e-4</sub>	1.64e-3 <sub>4.95e-4</sub>	6.41e-4 <sub>2.15e-4</sub>	1.30e-1 <sub>5.22e-2</sub>
DTLZ7	4.33e-2 <sub>8.34e-3</sub>	6.74e-2 <sub>1.67e-2</sub>	1.57e-1 <sub>3.47e-2</sub>	1.85e-1 <sub>6.23e-3</sub>	1.89e-1 <sub>8.60e-3</sub>	5.93e-2 <sub>2.26e-2</sub>	6.15e-2 <sub>2.13e-2</sub>	7.17e-2 <sub>2.40e-2</sub>	1.15e-1 <sub>3.31e-2</sub>	3.64e-1 <sub>3.79e-2</sub>
ZDT3	4.88e-3 <sub>1.03e-3</sub>	4.95e-3 <sub>1.03e-3</sub>	3.88e-3 <sub>8.47e-4</sub>	1.85e-3 <sub>9.00e-6</sub>	2.33e-1 <sub>1.20e-1</sub>	4.30e-3 <sub>8.50e-4</sub>	4.18e-3 <sub>9.31e-4</sub>	3.49e-3 <sub>8.22e-4</sub>	1.30e-3 <sub>3.41e-4</sub>	1.21e-1 <sub>4.82e-2</sub>

Table G.5.: Local search study – PR of CMA-ES. Median and IQR (as subscript) results for different step sizes  $\sigma$  across 100 runs. Best performances are colored in green, second-best performances in blue. The largest PRs are obtained for a step size of  $1e-2$ , which constitutes a good balance between a narrow and a broad search within the basin.

	Sum of objectives					Nash				
	1e-5	1e-4	1e-3	1e-2	1e-1	1e-5	1e-4	1e-3	1e-2	1e-1
	DEB2DK $k = 1$	1.00 <sub>0.00</sub>	0.50 <sub>0.33</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	0.33 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>1.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>
DEB2DK $k = 2$	1.00 <sub>0.25</sub>	0.75 <sub>0.25</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	0.50 <sub>0.00</sub>	0.50 <sub>0.50</sub>	0.50 <sub>0.50</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>
DEB2DK $k = 3$	0.80 <sub>0.40</sub>	0.80 <sub>0.20</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	0.60 <sub>0.00</sub>	0.67 <sub>0.33</sub>	0.67 <sub>0.33</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	0.33 <sub>0.00</sub>
DEB2DK $k = 4$	0.83 <sub>0.33</sub>	0.83 <sub>0.17</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.17</sub>	0.67 <sub>0.00</sub>	0.50 <sub>0.50</sub>	0.50 <sub>0.00</sub>	1.00 <sub>0.00</sub>	0.75 <sub>0.25</sub>	0.50 <sub>0.00</sub>
DEB3DK $k = 1$	0.00 <sub>0.00</sub>	0.00 <sub>0.20</sub>	0.40 <sub>0.20</sub>	1.00 <sub>0.20</sub>	0.80 <sub>0.00</sub>	0.00 <sub>0.00</sub>	0.00 <sub>0.00</sub>	0.00 <sub>1.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>
DEB3DK $k = 2$	0.00 <sub>0.10</sub>	0.00 <sub>0.10</sub>	0.40 <sub>0.20</sub>	0.80 <sub>0.10</sub>	0.50 <sub>0.10</sub>	0.00 <sub>0.00</sub>	0.00 <sub>0.00</sub>	0.25 <sub>0.25</sub>	0.75 <sub>0.50</sub>	0.50 <sub>0.25</sub>
DO2DK $k = 1 s = 1$	1.00 <sub>0.00</sub>	1.00 <sub>1.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>					
DO2DK $k = 2 s = 1$	1.00 <sub>0.50</sub>	0.50 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	0.50 <sub>0.00</sub>	1.00 <sub>0.00</sub>	0.50 <sub>0.50</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>
DO2DK $k = 3 s = 1$	1.00 <sub>0.00</sub>	1.00 <sub>0.50</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	0.50 <sub>0.50</sub>	1.00 <sub>0.00</sub>				
DO2DK $k = 4 s = 1$	1.00 <sub>0.33</sub>	0.33 <sub>0.33</sub>	1.00 <sub>0.33</sub>	1.00 <sub>0.33</sub>	0.33 <sub>0.00</sub>	1.00 <sub>0.50</sub>	0.50 <sub>0.50</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>
DTLZ7	0.00 <sub>0.00</sub>	0.00 <sub>0.00</sub>	0.50 <sub>0.25</sub>	0.75 <sub>0.25</sub>	0.50 <sub>0.50</sub>	0.00 <sub>0.00</sub>	0.00 <sub>0.00</sub>	0.00 <sub>0.25</sub>	0.25 <sub>0.50</sub>	0.25 <sub>0.25</sub>
ZDT3	1.00 <sub>0.00</sub>	0.60 <sub>0.20</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	0.80 <sub>0.20</sub>	0.00 <sub>0.00</sub>	0.00 <sub>0.20</sub>	0.80 <sub>0.00</sub>	0.80 <sub>0.00</sub>	0.40 <sub>0.20</sub>
	Angle utility					Tradeoff utility				
	1e-5	1e-4	1e-3	1e-2	1e-1	1e-5	1e-4	1e-3	1e-2	1e-1
	DEB2DK $k = 1$	1.00 <sub>0.75</sub>	1.00 <sub>0.00</sub>							
DEB2DK $k = 2$	0.50 <sub>0.50</sub>	0.50 <sub>0.50</sub>	1.00 <sub>0.00</sub>							
DEB2DK $k = 3$	0.33 <sub>0.33</sub>	0.67 <sub>0.33</sub>	1.00 <sub>0.00</sub>							
DEB2DK $k = 4$	0.50 <sub>0.50</sub>	0.75 <sub>0.25</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.25</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.11</sub>	1.00 <sub>0.11</sub>
DEB3DK $k = 1$	0.00 <sub>0.50</sub>	0.00 <sub>0.50</sub>	0.50 <sub>0.50</sub>	1.00 <sub>0.00</sub>	0.50 <sub>0.00</sub>	0.00 <sub>0.00</sub>	0.00 <sub>0.00</sub>	0.00 <sub>0.00</sub>	0.33 <sub>0.17</sub>	0.00 <sub>0.00</sub>
DEB3DK $k = 2$	0.00 <sub>0.00</sub>	0.00 <sub>0.04</sub>	0.04 <sub>0.04</sub>	0.00 <sub>0.00</sub>						
DO2DK $k = 1 s = 1$	1.00 <sub>1.00</sub>	0.50 <sub>1.00</sub>	1.00 <sub>0.00</sub>							
DO2DK $k = 2 s = 1$	1.00 <sub>0.00</sub>	0.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>							
DO2DK $k = 3 s = 1$	1.00 <sub>1.00</sub>	0.00 <sub>0.00</sub>	1.00 <sub>0.20</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>				
DO2DK $k = 4 s = 1$	1.00 <sub>1.00</sub>	0.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>							
DTLZ7	0.00 <sub>0.00</sub>	0.00 <sub>0.00</sub>	0.00 <sub>0.20</sub>	0.00 <sub>0.20</sub>	0.00 <sub>0.20</sub>	0.00 <sub>0.00</sub>	0.00 <sub>0.08</sub>	0.08 <sub>0.08</sub>	0.08 <sub>0.00</sub>	0.00 <sub>0.00</sub>
ZDT3	0.00 <sub>0.00</sub>	0.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	0.40 <sub>0.20</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.17</sub>	1.00 <sub>0.17</sub>

Table G.6.: Local search study – PR of the GA. Median and IQR results for different distribution indices  $\eta$  across 100 runs. Best performances are colored in green, second-best performances in blue. Largest PRs are obtained for a distribution index of 120. Smaller distribution indices broaden the search too far such that the GA escapes its assigned basin.

	Sum of objectives					Nash				
	20	30	40	80	120	20	30	40	80	120
DEB2DK $k = 1$	0.33 <sub>0.00</sub>	0.33 <sub>0.33</sub>	0.67 <sub>0.67</sub>	1.00 <sub>0.00</sub>						
DEB2DK $k = 2$	0.50 <sub>0.00</sub>	0.50 <sub>0.00</sub>	0.50 <sub>0.00</sub>	0.75 <sub>0.25</sub>	1.00 <sub>0.00</sub>					
DEB2DK $k = 3$	0.60 <sub>0.00</sub>	0.60 <sub>0.00</sub>	0.60 <sub>0.00</sub>	0.60 <sub>0.00</sub>	0.80 <sub>0.20</sub>	0.33 <sub>0.00</sub>	0.67 <sub>0.33</sub>	1.00 <sub>0.33</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>
DEB2DK $k = 4$	0.50 <sub>0.17</sub>	0.67 <sub>0.00</sub>	0.67 <sub>0.00</sub>	0.67 <sub>0.00</sub>	0.67 <sub>0.00</sub>	0.50 <sub>0.00</sub>	0.50 <sub>0.00</sub>	0.50 <sub>0.00</sub>	0.50 <sub>0.25</sub>	0.75 <sub>0.25</sub>
DEB3DK $k = 1$	0.20 <sub>0.20</sub>	0.60 <sub>0.20</sub>	0.80 <sub>0.20</sub>	0.80 <sub>0.20</sub>	0.80 <sub>0.20</sub>	1.00 <sub>1.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>
DEB3DK $k = 2$	0.10 <sub>0.20</sub>	0.40 <sub>0.20</sub>	0.55 <sub>0.30</sub>	0.80 <sub>0.10</sub>	0.80 <sub>0.20</sub>	0.00 <sub>0.25</sub>	0.25 <sub>0.25</sub>	0.50 <sub>0.25</sub>	0.75 <sub>0.00</sub>	0.75 <sub>0.25</sub>
DO2DK $k = 1 s = 1$	1.00 <sub>0.00</sub>									
DO2DK $k = 2 s = 1$	0.50 <sub>0.00</sub>	0.50 <sub>0.00</sub>	0.50 <sub>0.00</sub>	1.00 <sub>0.00</sub>						
DO2DK $k = 3 s = 1$	0.50 <sub>0.50</sub>	1.00 <sub>0.00</sub>								
DO2DK $k = 4 s = 1$	0.33 <sub>0.00</sub>	0.33 <sub>0.00</sub>	0.33 <sub>0.33</sub>	0.67 <sub>0.00</sub>	0.67 <sub>0.33</sub>	1.00 <sub>0.50</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>
DTLZ7	0.00 <sub>0.00</sub>	0.00 <sub>0.25</sub>	0.25 <sub>0.25</sub>	0.50 <sub>0.44</sub>	0.75 <sub>0.25</sub>	0.00 <sub>0.25</sub>	0.00 <sub>0.25</sub>	0.25 <sub>0.25</sub>	0.25 <sub>0.25</sub>	0.25 <sub>0.00</sub>
ZDT3	0.60 <sub>0.20</sub>	0.80 <sub>0.20</sub>	1.00 <sub>0.15</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	0.40 <sub>0.20</sub>	0.60 <sub>0.00</sub>	0.80 <sub>0.20</sub>	0.80 <sub>0.00</sub>	0.80 <sub>0.00</sub>
	Angle utility					Tradeoff utility				
	20	30	40	80	120	20	30	40	80	120
DEB2DK $k = 1$	1.00 <sub>0.00</sub>									
DEB2DK $k = 2$	1.00 <sub>0.00</sub>									
DEB2DK $k = 3$	1.00 <sub>0.33</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.14</sub>	1.00 <sub>0.11</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>
DEB2DK $k = 4$	0.75 <sub>0.25</sub>	1.00 <sub>0.25</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	0.89 <sub>0.19</sub>	0.89 <sub>0.11</sub>	0.89 <sub>0.11</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>
DEB3DK $k = 1$	0.50 <sub>0.00</sub>	1.00 <sub>0.50</sub>	1.00 <sub>0.50</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	0.00 <sub>0.17</sub>	0.17 <sub>0.29</sub>	0.17 <sub>0.17</sub>	0.33 <sub>0.17</sub>	0.17 <sub>0.17</sub>
DEB3DK $k = 2$	0.00 <sub>0.25</sub>	0.25 <sub>0.50</sub>	0.25 <sub>0.25</sub>	0.25 <sub>0.50</sub>	0.25 <sub>0.50</sub>	0.00 <sub>0.00</sub>	0.04 <sub>0.04</sub>	0.04 <sub>0.04</sub>	0.09 <sub>0.04</sub>	0.09 <sub>0.08</sub>
DO2DK $k = 1 s = 1$	1.00 <sub>0.00</sub>									
DO2DK $k = 2 s = 1$	1.00 <sub>0.00</sub>									
DO2DK $k = 3 s = 1$	1.00 <sub>0.00</sub>	1.00 <sub>0.20</sub>	1.00 <sub>0.20</sub>	1.00 <sub>0.20</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>				
DO2DK $k = 4 s = 1$	1.00 <sub>0.00</sub>	1.00 <sub>0.20</sub>	1.00 <sub>0.20</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>				
DTLZ7	0.00 <sub>0.00</sub>	0.00 <sub>0.00</sub>	0.00 <sub>0.00</sub>	0.00 <sub>0.00</sub>	0.00 <sub>0.20</sub>	0.00 <sub>0.00</sub>	0.08 <sub>0.08</sub>	0.17 <sub>0.17</sub>	0.25 <sub>0.17</sub>	0.25 <sub>0.17</sub>
ZDT3	0.60 <sub>0.20</sub>	1.00 <sub>0.20</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.17</sub>	0.83 <sub>0.17</sub>	0.83 <sub>0.17</sub>	0.83 <sub>0.00</sub>	0.83 <sub>0.17</sub>



Table G.8.: Local search study – PR of the PSA. Median and IQR (as subscript) results for different inertia weights  $w$  across 100 runs. Best performances are colored in green, second-best performances in blue. Largest PRs are obtained for an inertia weight of 0.75. Smaller inertia weights also achieve high PRs suggesting that the PSA is robust considering the chosen value for  $w$ . On the other hand, an inertia weight of  $w = 1$  results in erratic movements that makes the PSA leaving its assigned basin.

	Sum of objectives					Nash				
	0.1	0.25	0.5	0.75	1	0.1	0.25	0.5	0.75	1
DEB2DK $k = 1$	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	0.83 <sub>0.33</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>1.00</sub>
DEB2DK $k = 2$	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	0.75 <sub>0.25</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	0.50 <sub>0.00</sub>
DEB2DK $k = 3$	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	0.60 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	0.33 <sub>0.00</sub>
DEB2DK $k = 4$	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	0.67 <sub>0.17</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	0.75 <sub>0.25</sub>	0.50 <sub>0.00</sub>	0.50 <sub>0.25</sub>
DEB3DK $k = 1$	0.00 <sub>0.00</sub>	0.00 <sub>0.00</sub>	0.00 <sub>0.20</sub>	1.00 <sub>0.20</sub>	0.60 <sub>0.20</sub>	0.00 <sub>0.00</sub>	0.00 <sub>0.00</sub>	0.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	0.00 <sub>0.00</sub>
DEB3DK $k = 2$	0.00 <sub>0.00</sub>	0.00 <sub>0.00</sub>	0.00 <sub>0.10</sub>	0.90 <sub>0.10</sub>	0.40 <sub>0.00</sub>	0.00 <sub>0.00</sub>	0.00 <sub>0.00</sub>	0.00 <sub>0.00</sub>	0.75 <sub>0.25</sub>	0.00 <sub>0.25</sub>
DO2DK $k = 1 s = 1$	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>1.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>1.00</sub>
DO2DK $k = 2 s = 1$	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	0.50 <sub>0.00</sub>	0.50 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	0.50 <sub>0.00</sub>
DO2DK $k = 3 s = 1$	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	0.50 <sub>0.50</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>1.00</sub>
DO2DK $k = 4 s = 1$	1.00 <sub>0.33</sub>	1.00 <sub>0.33</sub>	0.67 <sub>0.00</sub>	0.67 <sub>0.00</sub>	0.33 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	0.50 <sub>0.00</sub>
DTLZ7	0.00 <sub>0.00</sub>	0.00 <sub>0.00</sub>	0.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	0.25 <sub>0.25</sub>	0.00 <sub>0.00</sub>	0.00 <sub>0.00</sub>	0.00 <sub>0.00</sub>	0.25 <sub>0.00</sub>	0.25 <sub>0.00</sub>
ZDT3	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	0.20 <sub>0.00</sub>	0.80 <sub>0.00</sub>	0.80 <sub>0.00</sub>	0.80 <sub>0.00</sub>	0.80 <sub>0.00</sub>	0.40 <sub>0.20</sub>

	Angle utility					Tradeoff utility				
	0.1	0.25	0.5	0.75	1	0.1	0.25	0.5	0.75	1
DEB2DK $k = 1$	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>1.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.33</sub>
DEB2DK $k = 2$	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	0.50 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	0.60 <sub>0.00</sub>
DEB2DK $k = 3$	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	0.33 <sub>0.25</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	0.57 <sub>0.14</sub>
DEB2DK $k = 4$	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	0.50 <sub>0.25</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	0.44 <sub>0.00</sub>
DEB3DK $k = 1$	0.00 <sub>0.00</sub>	0.00 <sub>0.00</sub>	0.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	0.50 <sub>0.50</sub>	0.00 <sub>0.00</sub>	0.00 <sub>0.00</sub>	0.00 <sub>0.00</sub>	0.25 <sub>0.17</sub>	0.33 <sub>0.17</sub>
DEB3DK $k = 2$	0.00 <sub>0.00</sub>	0.00 <sub>0.00</sub>	0.00 <sub>0.00</sub>	0.00 <sub>0.25</sub>	0.25 <sub>0.25</sub>	0.00 <sub>0.00</sub>	0.00 <sub>0.00</sub>	0.00 <sub>0.00</sub>	0.09 <sub>0.04</sub>	0.13 <sub>0.04</sub>
DO2DK $k = 1 s = 1$	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	0.00 <sub>1.00</sub>	1.00 <sub>0.00</sub>				
DO2DK $k = 2 s = 1$	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>1.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	0.67 <sub>0.00</sub>
DO2DK $k = 3 s = 1$	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	0.00 <sub>1.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	0.80 <sub>0.20</sub>	0.60 <sub>0.20</sub>
DO2DK $k = 4 s = 1$	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	0.00 <sub>1.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	0.40 <sub>0.20</sub>
DTLZ7	0.00 <sub>0.00</sub>	0.00 <sub>0.00</sub>	0.00 <sub>0.00</sub>	0.00 <sub>0.15</sub>	0.00 <sub>0.00</sub>	0.00 <sub>0.00</sub>	0.00 <sub>0.00</sub>	0.00 <sub>0.08</sub>	0.08 <sub>0.08</sub>	0.25 <sub>0.08</sub>
ZDT3	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	0.40 <sub>0.20</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	1.00 <sub>0.00</sub>	0.50 <sub>0.17</sub>

G. Statistics

Table G.9.: Local search study – FP of CMA-ES. Median results for different step sizes  $\sigma$  across 100 runs. Best performances are colored in green, second-best performances in blue. Choosing a step size of  $1e-2$  results in the least FPs. Smaller step sizes tend to narrow the search too much such that CMA-ES does not properly converge.

	Sum of objectives					Nash				
	1e-5	1e-4	1e-3	1e-2	1e-1	1e-5	1e-4	1e-3	1e-2	1e-1
DEB2DK $k = 1$	0	1	0	0	2	0	0	0	0	0
DEB2DK $k = 2$	0	1	0	0	2	1	1	0	0	0
DEB2DK $k = 3$	0	1	0	0	2	1	1	0	0	2
DEB2DK $k = 4$	1	1	0	0	2	2	2	0	0	2
DEB3DK $k = 1$	0	0	2	0	1	0	0	0	0	0
DEB3DK $k = 2$	0	0	2	0	4	0	0	3	0	0
DO2DK $k = 1 s = 1$	0	0	0	0	0	0	0	0	0	0
DO2DK $k = 2 s = 1$	0	1	0	0	1	0	1	0	0	0
DO2DK $k = 3 s = 1$	0	0	0	0	1	0	0	0	0	0
DO2DK $k = 4 s = 1$	0	1	0	0	1	0	0	0	0	0
DTLZ7	0	0	1	0	0	4	4	1	0	0
ZDT3	0	2	0	0	1	5	5	1	1	3

	Angle utility					Tradeoff utility				
	1e-5	1e-4	1e-3	1e-2	1e-1	1e-5	1e-4	1e-3	1e-2	1e-1
DEB2DK $k = 1$	0	0	0	0	0	0	0	0	0	0
DEB2DK $k = 2$	1	1	0	0	0	0	0	0	0	0
DEB2DK $k = 3$	1	1	0	0	0	0	0	0	0	0
DEB2DK $k = 4$	2	1	0	0	0	0	0	0	0	0
DEB3DK $k = 1$	0	0	0	0	1	0	0	0	0	0
DEB3DK $k = 2$	0	0	3	0	2	5	5	5	5.5	5
DO2DK $k = 1 s = 1$	0	0	0	0	0	0	0	0	0	0
DO2DK $k = 2 s = 1$	0	1	0	0	0	0	0	0	0	0
DO2DK $k = 3 s = 1$	0	1	0	0	0	0	0	0	0	0
DO2DK $k = 4 s = 1$	0	1	0	0	0	0	0	0	0	0
DTLZ7	5	4	2	2	3	1	1	2	4	1
ZDT3	5	5	0	0	3	0	0	0	0	0

Table G.10.: Local search study – FP of the GA. Median results for different distribution indices  $\eta$  across 100 runs. Best performances are colored in green, second-best performances in blue. The least number of FPs are obtained for a distribution index of 120. The results are mostly homogenous across the different scalarization functions and parametrizations suggesting that the GA converges well irrespective of the chosen distribution index.

	Sum of objectives					Nash				
	20	30	40	80	120	20	30	40	80	120
DEB2DK $k = 1$	2	2	1	0	0	0	0	0	0	0
DEB2DK $k = 2$	2	2	2	1	0	0	0	0	0	0
DEB2DK $k = 3$	2	2	2	2	1	2	1	0	0	0
DEB2DK $k = 4$	2	2	2	2	2	2	2	2	2	0
DEB3DK $k = 1$	1	1	1	0	0	0	0	0	0	0
DEB3DK $k = 2$	2	2	2	1	1	0	0	0	0	0
DO2DK $k = 1 s = 1$	0	0	0	0	0	0	0	0	0	0
DO2DK $k = 2 s = 1$	1	1	1	0	0	0	0	0	0	0
DO2DK $k = 3 s = 1$	1	0	0	0	0	0	0	0	0	0
DO2DK $k = 4 s = 1$	1	1	1	1	1	0	0	0	0	0
DTLZ7	0	0	0	0	0	0	0	0	0	0
ZDT3	1	1	0	0	0	3	2	1	1	1
	Angle utility					Tradeoff utility				
	20	30	40	80	120	20	30	40	80	120
DEB2DK $k = 1$	0	0	0	0	0	0	0	0	0	0
DEB2DK $k = 2$	0	0	0	0	0	0	0	0	0	0
DEB2DK $k = 3$	0	0	0	0	0	0	0	0	0	0
DEB2DK $k = 4$	1	0	0	0	0	0	0	0	0	0
DEB3DK $k = 1$	0	0	0	0	0	0	0	0	0	0
DEB3DK $k = 2$	0	0	0	0	0	10	9	7	6	6
DO2DK $k = 1 s = 1$	0	0	0	0	0	0	0	0	0	0
DO2DK $k = 2 s = 1$	0	0	0	0	0	0	0	0	0	0
DO2DK $k = 3 s = 1$	0	0	0	0	0	0	0	0	0	0
DO2DK $k = 4 s = 1$	0	0	0	0	0	0	0	0	0	0
DTLZ7	2	2	2	2	2	5	5	5	5	5
ZDT3	2	0	0	0	0	0	0	0	0	0

G. Statistics

Table G.11.: Local search study – FP of the HC. Median results for different distribution indices  $\eta$  across 100 runs. Best performances are colored in green, second-best performances in blue. The least number of FPs are obtained for a distribution index of 120.

	Sum of objectives					Nash				
	20	30	40	80	120	20	30	40	80	120
DEB2DK $k = 1$	2	2	1	0	0	0	0	0	0	0
DEB2DK $k = 2$	2	2	2	2	0	0	0	0	0	0
DEB2DK $k = 3$	2	2	2	2	2	2	1	1	0	0
DEB2DK $k = 4$	3	2	2	2	2	2	2	2	2	0
DEB3DK $k = 1$	1	1	1	0	0	0	0	0	0	0
DEB3DK $k = 2$	2	2	2	1	0	0	0	0	0	0
DO2DK $k = 1 s = 1$	0	0	0	0	0	0	0	0	0	0
DO2DK $k = 2 s = 1$	1	1	1	0	0	0	0	0	0	0
DO2DK $k = 3 s = 1$	1	0	0	0	0	0	0	0	0	0
DO2DK $k = 4 s = 1$	1	1	1	1	1	0	0	0	0	0
DTLZ7	0	0	0	0	0	1	0	0	0	0
ZDT3	2	1	0	0	0	3	2	1	1	1

	Angle utility					Tradeoff utility				
	20	30	40	80	120	20	30	40	80	120
DEB2DK $k = 1$	0	0	0	0	0	0	0	0	0	0
DEB2DK $k = 2$	0	0	0	0	0	1	1	0	0	0
DEB2DK $k = 3$	1	0	0	0	0	2	1	0	0	0
DEB2DK $k = 4$	2	0	0	0	0	2	2	1	0	0
DEB3DK $k = 1$	1	0	0	0	0	0	0	0	0	0
DEB3DK $k = 2$	0	0	0	0	0	14	11	9	6	6
DO2DK $k = 1 s = 1$	0	0	0	0	0	0	0	0	0	0
DO2DK $k = 2 s = 1$	0	0	0	0	0	1	0	0	0	0
DO2DK $k = 3 s = 1$	0	0	0	0	0	1	1	0	0	0
DO2DK $k = 4 s = 1$	0	0	0	0	0	1	1	1	0	0
DTLZ7	2	2	2	2	2	7	6	5	5	5
ZDT3	2	1	0	0	0	0	0	0	0	0

Table G.12.: Local search study – FP of the PSA. Median results for different inertia weights  $w$  across 100 runs. Best performances are colored in green, second-best performances in blue. The least number of FPs are obtained for inertia weights of 0.1 and 0.25. Larger inertia weights appear to impede the PSA to converge.

	Sum of objectives					Nash				
	0.1	0.25	0.5	0.75	1	0.1	0.25	0.5	0.75	1
DEB2DK $k = 1$	0	0	0	0	0	0	0	0	0	0
DEB2DK $k = 2$	0	0	0	0	0	0	0	0	0	1
DEB2DK $k = 3$	0	0	0	0	2	0	0	0	0	2
DEB2DK $k = 4$	0	0	0	0	2	0	0	1	2	2
DEB3DK $k = 1$	0	0	0	0	0	0	0	0	0	0
DEB3DK $k = 2$	0	0	0	0	6	0	0	0	0	2
DO2DK $k = 1$ $s = 1$	0	0	0	0	0	0	0	0	0	0
DO2DK $k = 2$ $s = 1$	0	0	0	1	1	0	0	0	0	1
DO2DK $k = 3$ $s = 1$	0	0	0	0	1	0	0	0	0	0
DO2DK $k = 4$ $s = 1$	0	0	1	1	1	0	0	0	0	0
DTLZ7	0	0	0	0	1	0	0	0	0	3
ZDT3	0	0	0	0	4	0	0	1	1	3

	Angle utility					Tradeoff utility				
	0.1	0.25	0.5	0.75	1	0.1	0.25	0.5	0.75	1
DEB2DK $k = 1$	0	0	0	0	0	0	0	0	0	0
DEB2DK $k = 2$	0	0	0	0	1	0	0	0	0	2
DEB2DK $k = 3$	0	0	0	0	2	0	0	0	0	2
DEB2DK $k = 4$	0	0	0	0	2	0	0	0	0	2
DEB3DK $k = 1$	0	0	0	0	0	0	0	0	0	0
DEB3DK $k = 2$	0	0	0	0	2	5	5	5	6	10
DO2DK $k = 1$ $s = 1$	0	0	0	0	0	0	0	0	0	0
DO2DK $k = 2$ $s = 1$	0	0	0	0	0	0	0	0	0	1
DO2DK $k = 3$ $s = 1$	0	0	0	0	0	0	0	0	0	1
DO2DK $k = 4$ $s = 1$	0	0	0	0	0	0	0	0	0	2
DTLZ7	0	0	2	2	2	1	1	2	4	7
ZDT3	0	0	0	0	3	0	0	0	0	2

Table G.13.: Cogeneration study – IGD. Median and IQR (as subscript) results after 50 000 function evaluations. Best performances are colored in green, second best in blue. ESPEA outperforms all other algorithms on each problem instance.

	ESPEA	IBEA	MOEAD	NSGA-II	NSGA-III	SMPSO	SMS-EMOA	SPEA2
CG0	2.87e-42, 77e-6	3.35e-49, 13e-8	4.29e-44, 71e-6	4.86e-45, 49e-5	4.38e-41, 11e-4	4.55e-45, 26e-5	4.80e-41, 29e-5	3.58e-43, 33e-5
CG1	1.69e-42, 18e-6	1.48e-31, 65e-8	1.47e-39, 95e-5	3.56e-46, 62e-5	1.20e-33, 18e-4	3.37e-45, 55e-5	1.49e-33, 87e-6	2.27e-43, 17e-5
CG2	1.70e-42, 34e-6	1.48e-31, 46e-8	1.47e-38, 81e-5	3.59e-45, 52e-5	1.15e-32, 87e-4	3.44e-45, 43e-5	1.49e-35, 43e-6	2.32e-43, 67e-5
CG3	1.70e-42, 17e-6	1.48e-31, 25e-8	1.47e-33, 50e-5	3.60e-46, 01e-5	1.14e-33, 07e-4	3.28e-44, 79e-5	1.49e-34, 32e-6	2.36e-43, 44e-5
CG4	1.70e-42, 21e-6	1.47e-31, 10e-8	1.46e-34, 57e-5	3.63e-46, 51e-5	1.20e-33, 56e-4	3.36e-45, 53e-5	1.47e-35, 27e-6	2.35e-43, 46e-5
CG5	1.69e-41, 82e-6	1.48e-31, 20e-8	1.47e-38, 08e-5	3.64e-45, 97e-5	1.15e-33, 14e-4	3.29e-44, 89e-5	1.49e-34, 36e-6	2.34e-42, 50e-5
CG6	1.70e-41, 97e-6	1.48e-31, 38e-8	1.46e-34, 80e-5	3.54e-46, 20e-5	1.18e-33, 12e-4	3.27e-44, 86e-5	1.48e-34, 87e-6	2.35e-43, 14e-5
CG7	1.69e-42, 39e-6	1.47e-31, 17e-8	1.46e-35, 64e-5	3.54e-46, 09e-5	1.21e-32, 65e-4	3.30e-44, 65e-5	1.48e-34, 92e-6	2.28e-42, 66e-5
CG8	1.70e-42, 35e-6	1.49e-31, 29e-8	1.47e-38, 80e-5	3.65e-45, 17e-5	1.26e-33, 11e-4	3.27e-45, 50e-5	1.49e-34, 77e-6	2.32e-43, 45e-5
CG9	1.69e-42, 18e-6	1.47e-31, 34e-8	1.46e-37, 09e-5	3.57e-46, 90e-5	1.23e-33, 12e-4	3.34e-45, 54e-5	1.48e-35, 25e-6	2.29e-42, 90e-5
CG10	1.70e-42, 51e-6	1.48e-31, 43e-8	1.47e-36, 65e-5	3.64e-46, 38e-5	1.19e-33, 10e-4	3.35e-45, 75e-5	1.49e-33, 370e-6	2.33e-43, 06e-5
CG11	1.69e-42, 10e-6	1.46e-31, 32e-8	1.45e-35, 62e-5	3.56e-46, 05e-5	1.23e-32, 89e-4	3.40e-44, 34e-5	1.47e-33, 80e-6	2.28e-43, 09e-5
CG12	1.70e-41, 78e-6	1.48e-31, 22e-8	1.46e-38, 00e-5	3.75e-46, 61e-5	1.19e-32, 51e-4	3.38e-46, 37e-5	1.48e-34, 72e-6	2.33e-43, 77e-5
CG13	1.68e-42, 68e-6	1.48e-31, 39e-8	1.47e-38, 51e-5	3.68e-46, 47e-5	1.21e-33, 06e-4	3.32e-44, 45e-5	1.49e-34, 38e-6	2.29e-43, 51e-5
CG14	1.69e-42, 37e-6	1.48e-31, 32e-8	1.46e-37, 76e-5	3.60e-46, 52e-5	1.20e-32, 88e-4	3.32e-44, 20e-5	1.49e-34, 01e-6	2.35e-43, 07e-5
CG15	1.70e-42, 41e-6	1.49e-31, 08e-8	1.46e-35, 40e-5	3.51e-45, 99e-5	1.19e-33, 24e-4	3.30e-44, 90e-5	1.49e-34, 92e-6	2.29e-43, 69e-5
CG16	1.69e-42, 46e-6	1.48e-31, 47e-8	1.46e-36, 26e-5	3.52e-45, 45e-5	1.22e-32, 48e-4	3.38e-45, 94e-5	1.49e-34, 32e-6	2.30e-43, 75e-5
CG17	1.70e-41, 98e-6	1.48e-39, 46e-9	1.47e-35, 62e-5	3.60e-45, 87e-5	1.17e-33, 35e-4	3.36e-45, 18e-5	1.49e-34, 61e-6	2.34e-43, 69e-5
CG18	1.68e-42, 54e-6	1.48e-31, 14e-8	1.46e-39, 28e-5	3.58e-45, 72e-5	1.21e-33, 51e-4	3.35e-44, 25e-5	1.48e-34, 18e-6	2.30e-43, 70e-5
CG19	1.69e-42, 01e-6	1.48e-31, 31e-8	1.46e-37, 09e-5	3.56e-45, 09e-5	1.24e-33, 15e-4	3.37e-45, 27e-5	1.49e-35, 09e-6	2.31e-42, 66e-5
CG20	1.69e-42, 55e-6	1.48e-31, 42e-8	1.47e-37, 50e-5	3.59e-45, 60e-5	1.21e-32, 99e-4	3.41e-44, 35e-5	1.49e-34, 32e-6	2.32e-42, 41e-5
CG21	1.69e-42, 83e-6	1.49e-31, 34e-8	1.48e-34, 40e-5	3.56e-45, 35e-5	1.22e-33, 21e-4	3.34e-44, 68e-5	1.49e-33, 85e-6	2.30e-43, 32e-5
CG22	1.69e-42, 53e-6	1.47e-31, 34e-8	1.45e-39, 30e-5	3.59e-44, 77e-5	1.18e-33, 51e-4	3.33e-44, 67e-5	1.48e-35, 10e-6	2.27e-44, 19e-5
CG23	1.69e-42, 34e-6	1.47e-31, 66e-8	1.46e-34, 92e-5	3.58e-45, 35e-5	1.22e-32, 49e-4	3.36e-44, 95e-5	1.48e-34, 51e-6	2.28e-42, 73e-5
CG24	1.69e-41, 91e-6	1.48e-31, 56e-8	1.46e-31, 05e-4	3.57e-46, 96e-5	1.21e-32, 69e-4	3.39e-45, 93e-5	1.49e-34, 80e-6	2.38e-43, 33e-5
CG25	1.68e-42, 70e-6	1.48e-31, 20e-8	1.47e-36, 73e-5	3.57e-46, 51e-5	1.23e-32, 88e-4	3.32e-44, 46e-5	1.49e-34, 36e-6	2.30e-42, 73e-5
CG26	1.69e-42, 63e-6	1.47e-31, 55e-8	1.46e-38, 10e-5	3.65e-46, 65e-5	1.16e-33, 35e-4	3.39e-44, 54e-5	1.48e-35, 28e-6	2.33e-42, 92e-5
CG27	1.70e-42, 59e-6	1.48e-31, 47e-8	1.46e-37, 73e-5	3.57e-46, 68e-5	1.24e-32, 99e-4	3.45e-45, 05e-5	1.49e-35, 10e-6	2.31e-43, 26e-5
CG28	1.68e-42, 02e-6	1.46e-31, 30e-8	1.44e-38, 23e-5	3.57e-45, 21e-5	1.25e-32, 94e-4	3.31e-46, 50e-5	1.47e-34, 38e-6	2.32e-43, 30e-5
CG29	1.69e-42, 09e-6	1.47e-31, 46e-8	1.46e-37, 01e-5	3.61e-46, 37e-5	1.23e-32, 28e-4	3.32e-45, 80e-5	1.48e-33, 82e-6	2.33e-43, 44e-5
CG30	1.70e-42, 40e-6	1.48e-31, 32e-8	1.46e-34, 67e-5	3.70e-45, 12e-5	1.17e-33, 15e-4	3.36e-46, 38e-5	1.48e-35, 20e-6	2.35e-43, 47e-5
CG31	1.70e-42, 37e-6	1.48e-31, 26e-8	1.47e-33, 82e-5	3.69e-46, 17e-5	1.17e-33, 52e-4	3.38e-44, 97e-5	1.48e-34, 78e-6	2.29e-43, 86e-5
CG32	1.67e-42, 17e-6	1.48e-31, 35e-8	1.46e-39, 80e-5	3.72e-45, 83e-5	1.22e-32, 61e-4	3.34e-44, 96e-5	1.48e-34, 53e-6	2.35e-43, 97e-5
CG33	1.68e-42, 68e-6	1.47e-31, 29e-8	1.45e-35, 32e-5	3.53e-47, 03e-5	1.20e-32, 84e-4	3.32e-45, 00e-5	1.47e-35, 27e-6	2.32e-42, 67e-5
CG34	1.69e-42, 32e-6	1.48e-31, 47e-8	1.47e-34, 86e-5	3.63e-45, 68e-5	1.16e-32, 44e-4	3.39e-46, 54e-5	1.48e-33, 80e-6	2.34e-43, 29e-5
CG35	1.69e-41, 96e-6	1.47e-31, 32e-8	1.46e-33, 38e-5	3.60e-46, 89e-5	1.20e-32, 22e-4	3.41e-44, 30e-5	1.47e-34, 96e-6	2.31e-43, 24e-5
CG36	1.69e-42, 04e-6	1.47e-31, 14e-8	1.46e-34, 47e-5	3.56e-44, 62e-5	1.20e-32, 55e-4	3.35e-44, 95e-5	1.47e-34, 08e-6	2.31e-43, 28e-5
CG37	1.69e-42, 04e-6	1.48e-31, 11e-8	1.46e-39, 72e-5	3.70e-45, 94e-5	1.25e-33, 35e-4	3.46e-46, 08e-5	1.48e-34, 05e-6	2.30e-43, 29e-5
CG38	1.69e-42, 40e-6	1.49e-39, 96e-9	1.48e-36, 72e-5	3.58e-46, 33e-5	1.22e-33, 54e-4	3.41e-44, 39e-5	1.49e-34, 73e-6	2.27e-42, 53e-5

Table G.14.: Cogeneration study – fuel efficiency. Median and IQR (as subscript) results after 50 000 function evaluations. Best performances are colored in green, second best in blue. Results are rounded to three significant digits. Fuel efficiencies are highly similar across the algorithms and problem instances.

	ESPEA DE	IBEA	MOEAD	NSGA-II	NSGA-III	SMP SO	SMS-EMOA	SPEA2
CG0	70.560 <sub>8.351e-5</sub>	70.559 <sub>0.001</sub>	70.560 <sub>1.588e-6</sub>	70.477 <sub>0.136</sub>	70.492 <sub>0.067</sub>	70.560 <sub>0.000</sub>	70.812 <sub>0.025</sub>	70.297 <sub>0.221</sub>
CG1	70.560 <sub>1.065e-4</sub>	70.560 <sub>8.843e-4</sub>	70.560 <sub>9.220e-7</sub>	70.493 <sub>0.125</sub>	70.502 <sub>0.042</sub>	70.560 <sub>0.000</sub>	70.538 <sub>0.029</sub>	70.264 <sub>0.278</sub>
CG2	70.560 <sub>1.259e-4</sub>	70.560 <sub>9.916e-4</sub>	70.560 <sub>1.241e-6</sub>	70.471 <sub>0.168</sub>	70.507 <sub>0.064</sub>	70.560 <sub>0.000</sub>	70.535 <sub>0.027</sub>	70.246 <sub>0.187</sub>
CG3	70.560 <sub>2.470e-4</sub>	70.560 <sub>0.002</sub>	70.560 <sub>1.279e-6</sub>	70.489 <sub>0.127</sub>	70.507 <sub>0.041</sub>	70.560 <sub>0.000</sub>	70.540 <sub>0.030</sub>	70.207 <sub>0.229</sub>
CG4	70.560 <sub>1.121e-4</sub>	70.560 <sub>0.001</sub>	70.560 <sub>1.306e-6</sub>	70.475 <sub>0.154</sub>	70.507 <sub>0.047</sub>	70.560 <sub>0.000</sub>	70.540 <sub>0.028</sub>	70.229 <sub>0.261</sub>
CG5	70.560 <sub>9.279e-5</sub>	70.560 <sub>0.001</sub>	70.560 <sub>1.332e-6</sub>	70.493 <sub>0.123</sub>	70.510 <sub>0.049</sub>	70.560 <sub>0.000</sub>	70.538 <sub>0.027</sub>	70.248 <sub>0.256</sub>
CG6	70.560 <sub>1.535e-4</sub>	70.560 <sub>0.001</sub>	70.560 <sub>1.444e-6</sub>	70.473 <sub>0.149</sub>	70.509 <sub>0.046</sub>	70.560 <sub>0.000</sub>	70.534 <sub>0.037</sub>	70.253 <sub>0.261</sub>
CG7	70.560 <sub>1.081e-4</sub>	70.560 <sub>0.001</sub>	70.560 <sub>1.263e-6</sub>	70.499 <sub>0.121</sub>	70.502 <sub>0.062</sub>	70.560 <sub>0.000</sub>	70.538 <sub>0.031</sub>	70.297 <sub>0.240</sub>
CG8	70.560 <sub>2.722e-4</sub>	70.560 <sub>9.587e-4</sub>	70.560 <sub>1.406e-6</sub>	70.448 <sub>0.187</sub>	70.499 <sub>0.061</sub>	70.560 <sub>0.000</sub>	70.540 <sub>0.032</sub>	70.267 <sub>0.257</sub>
CG9	70.560 <sub>8.413e-5</sub>	70.560 <sub>0.001</sub>	70.560 <sub>1.281e-6</sub>	70.467 <sub>0.159</sub>	70.503 <sub>0.057</sub>	70.560 <sub>0.000</sub>	70.541 <sub>0.025</sub>	70.249 <sub>0.185</sub>
CG10	70.560 <sub>2.290e-4</sub>	70.560 <sub>0.001</sub>	70.560 <sub>1.403e-6</sub>	70.482 <sub>0.152</sub>	70.502 <sub>0.051</sub>	70.560 <sub>0.000</sub>	70.538 <sub>0.025</sub>	70.234 <sub>0.242</sub>
CG11	70.560 <sub>1.309e-4</sub>	70.560 <sub>8.668e-4</sub>	70.560 <sub>1.552e-6</sub>	70.494 <sub>0.147</sub>	70.506 <sub>0.056</sub>	70.560 <sub>0.000</sub>	70.537 <sub>0.032</sub>	70.256 <sub>0.223</sub>
CG12	70.560 <sub>1.646e-4</sub>	70.560 <sub>0.001</sub>	70.560 <sub>1.224e-6</sub>	70.472 <sub>0.151</sub>	70.507 <sub>0.049</sub>	70.560 <sub>0.000</sub>	70.538 <sub>0.023</sub>	70.261 <sub>0.294</sub>
CG13	70.560 <sub>1.291e-4</sub>	70.560 <sub>6.912e-4</sub>	70.560 <sub>1.540e-6</sub>	70.473 <sub>0.113</sub>	70.507 <sub>0.055</sub>	70.560 <sub>0.000</sub>	70.538 <sub>0.023</sub>	70.263 <sub>0.279</sub>
CG14	70.560 <sub>2.248e-4</sub>	70.560 <sub>0.001</sub>	70.560 <sub>1.422e-6</sub>	70.474 <sub>0.139</sub>	70.493 <sub>0.060</sub>	70.560 <sub>0.000</sub>	70.543 <sub>0.029</sub>	70.228 <sub>0.254</sub>
CG15	70.560 <sub>2.952e-4</sub>	70.560 <sub>0.001</sub>	70.560 <sub>1.344e-6</sub>	70.481 <sub>0.122</sub>	70.497 <sub>0.050</sub>	70.560 <sub>0.000</sub>	70.538 <sub>0.028</sub>	70.247 <sub>0.234</sub>
CG16	70.560 <sub>1.442e-4</sub>	70.560 <sub>6.398e-4</sub>	70.560 <sub>1.466e-6</sub>	70.498 <sub>0.119</sub>	70.506 <sub>0.052</sub>	70.560 <sub>0.000</sub>	70.537 <sub>0.030</sub>	70.283 <sub>0.235</sub>
CG17	70.560 <sub>3.216e-4</sub>	70.560 <sub>9.659e-4</sub>	70.560 <sub>1.238e-6</sub>	70.476 <sub>0.122</sub>	70.504 <sub>0.053</sub>	70.560 <sub>0.000</sub>	70.541 <sub>0.030</sub>	70.254 <sub>0.246</sub>
CG18	70.560 <sub>1.545e-4</sub>	70.560 <sub>0.001</sub>	70.560 <sub>1.422e-6</sub>	70.465 <sub>0.121</sub>	70.502 <sub>0.053</sub>	70.560 <sub>0.000</sub>	70.541 <sub>0.024</sub>	70.233 <sub>0.248</sub>
CG19	70.560 <sub>1.131e-4</sub>	70.560 <sub>8.269e-4</sub>	70.560 <sub>9.418e-7</sub>	70.490 <sub>0.135</sub>	70.507 <sub>0.050</sub>	70.560 <sub>0.000</sub>	70.541 <sub>0.025</sub>	70.250 <sub>0.225</sub>
CG20	70.560 <sub>1.253e-4</sub>	70.560 <sub>0.001</sub>	70.560 <sub>1.314e-6</sub>	70.499 <sub>0.107</sub>	70.502 <sub>0.053</sub>	70.560 <sub>0.000</sub>	70.538 <sub>0.030</sub>	70.266 <sub>0.225</sub>
CG21	70.560 <sub>9.220e-5</sub>	70.560 <sub>0.002</sub>	70.560 <sub>1.219e-6</sub>	70.489 <sub>0.167</sub>	70.506 <sub>0.041</sub>	70.560 <sub>0.000</sub>	70.542 <sub>0.026</sub>	70.274 <sub>0.204</sub>
CG22	70.560 <sub>1.224e-4</sub>	70.560 <sub>0.001</sub>	70.560 <sub>1.427e-6</sub>	70.461 <sub>0.183</sub>	70.493 <sub>0.065</sub>	70.560 <sub>0.000</sub>	70.539 <sub>0.031</sub>	70.279 <sub>0.225</sub>
CG23	70.560 <sub>4.028e-4</sub>	70.560 <sub>9.208e-4</sub>	70.560 <sub>1.045e-6</sub>	70.462 <sub>0.164</sub>	70.510 <sub>0.067</sub>	70.560 <sub>0.000</sub>	70.537 <sub>0.026</sub>	70.244 <sub>0.241</sub>
CG24	70.560 <sub>6.275e-5</sub>	70.560 <sub>6.531e-4</sub>	70.560 <sub>1.162e-6</sub>	70.481 <sub>0.154</sub>	70.503 <sub>0.052</sub>	70.560 <sub>0.000</sub>	70.539 <sub>0.030</sub>	70.241 <sub>0.271</sub>
CG25	70.560 <sub>2.581e-4</sub>	70.560 <sub>9.179e-4</sub>	70.560 <sub>1.390e-6</sub>	70.487 <sub>0.092</sub>	70.499 <sub>0.047</sub>	70.560 <sub>0.000</sub>	70.540 <sub>0.025</sub>	70.258 <sub>0.300</sub>
CG26	70.560 <sub>1.073e-4</sub>	70.560 <sub>0.001</sub>	70.560 <sub>1.393e-6</sub>	70.486 <sub>0.145</sub>	70.503 <sub>0.039</sub>	70.560 <sub>0.000</sub>	70.540 <sub>0.024</sub>	70.237 <sub>0.202</sub>
CG27	70.560 <sub>2.373e-4</sub>	70.560 <sub>9.735e-4</sub>	70.560 <sub>1.407e-6</sub>	70.485 <sub>0.195</sub>	70.502 <sub>0.042</sub>	70.560 <sub>0.000</sub>	70.542 <sub>0.027</sub>	70.262 <sub>0.268</sub>
CG28	70.560 <sub>9.731e-5</sub>	70.560 <sub>0.001</sub>	70.560 <sub>1.355e-6</sub>	70.466 <sub>0.142</sub>	70.509 <sub>0.042</sub>	70.560 <sub>0.000</sub>	70.534 <sub>0.029</sub>	70.256 <sub>0.237</sub>
CG29	70.560 <sub>1.434e-4</sub>	70.560 <sub>0.001</sub>	70.560 <sub>1.294e-6</sub>	70.493 <sub>0.087</sub>	70.505 <sub>0.054</sub>	70.560 <sub>0.000</sub>	70.539 <sub>0.025</sub>	70.250 <sub>0.242</sub>
CG30	70.560 <sub>1.337e-4</sub>	70.560 <sub>0.001</sub>	70.560 <sub>1.370e-6</sub>	70.481 <sub>0.167</sub>	70.507 <sub>0.051</sub>	70.560 <sub>0.000</sub>	70.544 <sub>0.026</sub>	70.233 <sub>0.242</sub>
CG31	70.560 <sub>1.550e-4</sub>	70.560 <sub>0.001</sub>	70.560 <sub>1.513e-6</sub>	70.489 <sub>0.159</sub>	70.504 <sub>0.044</sub>	70.560 <sub>0.000</sub>	70.539 <sub>0.029</sub>	70.248 <sub>0.251</sub>
CG32	70.560 <sub>8.438e-5</sub>	70.560 <sub>0.001</sub>	70.560 <sub>1.059e-6</sub>	70.480 <sub>0.120</sub>	70.503 <sub>0.047</sub>	70.560 <sub>0.000</sub>	70.540 <sub>0.031</sub>	70.230 <sub>0.252</sub>
CG33	70.560 <sub>2.003e-4</sub>	70.560 <sub>8.869e-4</sub>	70.560 <sub>1.409e-6</sub>	70.484 <sub>0.143</sub>	70.498 <sub>0.053</sub>	70.560 <sub>0.000</sub>	70.537 <sub>0.033</sub>	70.232 <sub>0.265</sub>
CG34	70.560 <sub>2.688e-4</sub>	70.560 <sub>0.001</sub>	70.560 <sub>1.371e-6</sub>	70.476 <sub>0.147</sub>	70.499 <sub>0.048</sub>	70.560 <sub>0.000</sub>	70.535 <sub>0.025</sub>	70.261 <sub>0.227</sub>
CG35	70.560 <sub>9.429e-5</sub>	70.560 <sub>9.146e-4</sub>	70.560 <sub>1.389e-6</sub>	70.477 <sub>0.168</sub>	70.497 <sub>0.053</sub>	70.560 <sub>0.000</sub>	70.538 <sub>0.024</sub>	70.243 <sub>0.231</sub>
CG36	70.560 <sub>5.714e-5</sub>	70.560 <sub>0.001</sub>	70.560 <sub>1.167e-6</sub>	70.486 <sub>0.170</sub>	70.506 <sub>0.042</sub>	70.560 <sub>0.000</sub>	70.540 <sub>0.036</sub>	70.236 <sub>0.260</sub>
CG37	70.560 <sub>9.181e-5</sub>	70.560 <sub>9.653e-4</sub>	70.560 <sub>1.615e-6</sub>	70.477 <sub>0.159</sub>	70.508 <sub>0.056</sub>	70.560 <sub>0.000</sub>	70.541 <sub>0.023</sub>	70.252 <sub>0.278</sub>
CG38	70.560 <sub>1.513e-4</sub>	70.560 <sub>0.002</sub>	70.560 <sub>1.145e-6</sub>	70.486 <sub>0.130</sub>	70.503 <sub>0.067</sub>	70.560 <sub>0.000</sub>	70.538 <sub>0.025</sub>	70.265 <sub>0.234</sub>

*G. Statistics*

## **G.2. Normal Distribution and Group Differences**

Table G.15.: Comparative study of ESPEA - IGD. The table reports  $p$ -values of the Anderson-Darling and Kruskal-Wallis tests. Significant results at a 5 % level for the Kruskal-Wallis test are highlighted in green.

	Anderson-Darling								Kruskal-Wallis
	ESPEA	IBEA	MOEAD	NSGA-II	NSGA-III	SMPSO	SMS-EMOA	SPEA2	
B1	0.6424	0.7638	0.1660	0.8433	0.0593	0.0810	0.0005	0.6305	0.0000
B2	0.0287	0.0734	0.0026	0.0532	0.5539	0.6786	0.0005	0.2897	0.0000
B3	0.3559	0.2339	0.0005	0.7626	0.0012	0.0026	0.5145	0.3879	0.0000
B4	0.1582	0.1883	0.6124	0.0339	0.5122	0.0005	0.0005	0.7094	0.0000
B5	0.6799	0.4108	0.0005	0.0253	0.0118	0.6566	0.2506	0.1707	0.0000
B6	0.3496	0.5024	0.0260	0.0215	0.2224	0.0502	0.8920	0.0005	0.0000
DEB2DK $k = 1$	0.0012	0.2541	0.0005	0.0021	0.0005	0.1413	0.3059	0.0031	0.0000
DEB2DK $k = 3$	0.0005	0.0005	0.0005	0.0029	0.0005	0.5033	0.7275	0.0271	0.0000
DEB3DK $k = 1$	0.3693	0.0918	0.0005	0.0005	0.4095	0.0005	0.8341	0.9077	0.0000
DEB3DK $k = 2$	0.8975	0.3051	0.0005	0.3243	0.0111	0.0005	0.0005	0.2323	0.0000
DO2DK $k = 2 \ s = 1$	0.0005	0.0005	0.0085	0.0005	0.0005	0.1101	0.3585	0.8467	0.0000
DO2DK $k = 4 \ s = 1$	0.0005	0.0005	0.0764	0.0310	0.0005	0.0005	0.9319	0.0145	0.0000
DTLZ1	0.0005	0.0918	0.2768	0.0005	0.0005	0.0226	0.7328	0.0089	0.0000
DTLZ3	0.0005	0.0005	0.0005	0.0005	0.0005	0.0005	0.0005	0.0005	0.0000
DTLZ5	0.0005	0.2209	0.7028	0.1786	0.7349	0.0020	0.2464	0.3308	0.0000
DTLZ7	0.0005	0.0005	0.0005	0.0005	0.0005	0.0023	0.0005	0.0005	0.0000
invDTLZ1	0.6547	0.0005	0.0053	0.0005	0.5033	0.2472	0.0005	0.0019	0.0000
Lamé $m = 2 \ \gamma = 0.25$	0.0236	0.0005	0.0005	0.0013	0.0005	0.2172	0.3021	0.0351	0.0000
Lamé $m = 3 \ \gamma = 0.5$	0.9880	0.0005	0.1869	0.0107	0.0005	0.0756	0.8861	0.0031	0.0000
Lamé $m = 2 \ \gamma = 4$	0.0971	0.7606	0.3616	0.0009	0.0005	0.3537	0.6047	0.0653	0.0000
ZDT1	0.0005	0.6084	0.0005	0.5585	0.0005	0.0592	0.0005	0.0446	0.0000
ZDT2	0.0005	0.0123	0.0005	0.0080	0.0005	0.3832	0.8269	0.4332	0.0000
ZDT3	0.3637	0.0005	0.0076	0.0005	0.0005	0.1839	0.0005	0.0565	0.0000
ZDT6	0.0005	0.0055	0.0373	0.0005	0.0005	0.3906	0.0005	0.0017	0.0000

## G. Statistics

Table G.16.: Comparative study of ESPEA – IGD. The table shows  $p$ -values of the Anderson-Darling and Kruskal-Wallis tests based on the grouping by Pareto front curvature. Significant results at a 5 % level for the Kruskal-Wallis test are highlighted in green. Significant performance differences are observed on all problem instances.

	Anderson-Darling								Kruskal-Wallis
	ESPEA	IBEA	MOEAD	NSGA-II	NSGA-III	SMPSO	SMS-EMOA	SPEA2	
Convex	0.0005	0.0005	0.0005	0.0026	0.0005	0.0005	0.0005	0.0005	0.0000
Concave	0.0069	0.0215	0.0049	0.1710	0.1934	0.0005	0.0006	0.0173	0.0000
Multimodal	0.2735	0.0005	0.0008	0.0096	0.0005	0.5969	0.0176	0.5258	0.0000

Table G.17.: Comparative study of ESPEA – spread. The table shows  $p$ -values of the Anderson-Darling and Kruskal-Wallis tests based on the grouping by Pareto front curvature. Significant results at a 5 % level for the Kruskal-Wallis test are highlighted in green. Significant performance differences are observed on all problem instances.

	Anderson-Darling								Kruskal-Wallis
	ESPEA	IBEA	MOEAD	NSGA-II	NSGA-III	SMPSO	SMS-EMOA	SPEA2	
Convex	0.5007	0.0005	0.0005	0.7835	0.0005	0.0005	0.0005	0.0005	0.0000
Concave	0.2348	0.0006	0.0005	0.0713	0.0005	0.0005	0.0056	0.0005	0.0000
Multimodal	0.4679	0.0005	0.0759	0.2789	0.0005	0.7207	0.4314	0.2010	0.0000

## G.2. Normal Distribution and Group Differences

Table G.18.: Local search study – PD. The table shows  $p$ -values of the Anderson-Darling and ANOVA/Kruskal-Wallis tests. Significant results at a 5% level for the ANOVA/Kruskal-Wallis tests are highlighted in green. Significant performance differences are observed on all problem instances.

	Anderson-Darling					Bartlett	ANOVA / Kruskal-Wallis
	Baseline	CMA-ES	GA	HC	PSA		
Sum of objectives							
DEB2DK $k = 1$	0.0005	0.0005	0.0005	0.0005	0.0682	n/a	0.0000
DEB2DK $k = 2$	0.1916	0.0083	0.0005	0.0005	0.0577	n/a	0.0000
DEB2DK $k = 3$	0.0519	0.0944	0.0005	0.0005	0.0005	n/a	0.0000
DEB2DK $k = 4$	0.0802	0.0240	0.0005	0.0005	0.0005	n/a	0.0000
DEB3DK $k = 1$	0.0005	0.0007	0.0005	0.0005	0.0005	n/a	0.0000
DEB3DK $k = 2$	0.0005	0.0642	0.0005	0.0005	0.0005	n/a	0.0000
DO2DK $k = 1 s = 1$	0.0005	0.0005	0.0005	0.0005	0.5847	n/a	0.0000
DO2DK $k = 2 s = 1$	0.0005	0.0005	0.0005	0.0005	0.0005	n/a	0.0000
DO2DK $k = 3 s = 1$	0.0208	0.0131	0.0005	0.0005	0.0005	n/a	0.0000
DO2DK $k = 4 s = 1$	0.0005	0.0005	0.0005	0.0005	0.0005	n/a	0.0007
DTLZ7	0.0228	0.0005	0.0585	0.8327	0.0005	n/a	0.0000
ZDT3	0.8949	0.0005	0.1245	0.0072	0.0005	n/a	0.0000
Nash							
DEB2DK $k = 1$	0.0005	0.0005	0.0005	0.0005	0.0005	n/a	0.0000
DEB2DK $k = 2$	0.0192	0.0098	0.0066	0.0005	0.7460	n/a	0.0000
DEB2DK $k = 3$	0.1566	0.0005	0.0656	0.0094	0.0382	n/a	0.0000
DEB2DK $k = 4$	0.0005	0.0005	0.0005	0.0005	0.0005	n/a	0.0000
DEB3DK $k = 1$	0.0113	0.0005	0.0974	0.0391	0.0005	n/a	0.0000
DEB3DK $k = 2$	0.0005	0.0005	0.0005	0.0005	0.0005	n/a	0.0000
DO2DK $k = 1 s = 1$	0.0005	0.0005	0.0005	0.0005	0.0005	n/a	0.0000
DO2DK $k = 2 s = 1$	0.0005	0.0005	0.0010	0.0005	0.0231	n/a	0.0000
DO2DK $k = 3 s = 1$	0.0005	0.0080	0.0005	0.0005	0.3373	n/a	0.0000
DO2DK $k = 4 s = 1$	0.0439	0.0149	0.1151	0.0005	0.0005	n/a	0.0000
DTLZ7	0.0005	0.0005	0.0005	0.0005	0.0005	n/a	0.0000
ZDT3	0.1758	0.3964	0.0639	0.5905	0.3955	0.0005	0.0000
Angle utility							
DEB2DK $k = 1$	0.0005	0.0005	0.0005	0.0005	0.0005	n/a	0.0000
DEB2DK $k = 2$	0.0117	0.0005	0.0012	0.0102	0.7129	n/a	0.0000
DEB2DK $k = 3$	0.0031	0.0725	0.0023	0.0005	0.0005	n/a	0.0000
DEB2DK $k = 4$	0.0005	0.0014	0.5948	0.1115	0.5757	n/a	0.0000
DEB3DK $k = 1$	0.0005	0.0005	0.0005	0.0111	0.0005	n/a	0.0000
DEB3DK $k = 2$	0.0005	0.0005	0.0005	0.0005	0.0005	n/a	0.0000
DO2DK $k = 1 s = 1$	0.0005	0.0005	0.0005	0.0005	0.0005	n/a	0.0000
DO2DK $k = 2 s = 1$	0.0005	0.0005	0.1513	0.0029	0.0005	n/a	0.0000
DO2DK $k = 3 s = 1$	0.0190	0.0005	0.0273	0.0262	0.0005	n/a	0.0000
DO2DK $k = 4 s = 1$	0.0005	0.1147	0.0005	0.0053	0.0011	n/a	0.0000
DTLZ7	0.0005	0.0005	0.0005	0.0005	0.0005	n/a	0.0000
ZDT3	0.8731	0.0116	0.5389	0.1182	0.0005	n/a	0.0000
Tradeoff utility							
DEB2DK $k = 1$	0.0005	0.0005	0.0005	0.0005	0.0005	n/a	0.0000
DEB2DK $k = 2$	0.4330	0.0005	0.0028	0.2696	0.0005	n/a	0.0000
DEB2DK $k = 3$	0.0008	0.0005	0.0418	0.0005	0.0005	n/a	0.0000
DEB2DK $k = 4$	0.6378	0.0079	0.6759	0.0005	0.0005	n/a	0.0000
DEB3DK $k = 1$	0.0052	0.1303	0.0005	0.0780	0.0965	n/a	0.0000
DEB3DK $k = 2$	0.3292	0.4768	0.7412	0.7958	0.7087	0.5193	0.0037
DO2DK $k = 1 s = 1$	0.0005	0.0005	0.0005	0.0005	0.0005	n/a	0.0000
DO2DK $k = 2 s = 1$	0.0165	0.4309	0.0005	0.1268	0.4248	n/a	0.0000
DO2DK $k = 3 s = 1$	0.0005	0.0005	0.0005	0.0217	0.0005	n/a	0.0000
DO2DK $k = 4 s = 1$	0.0463	0.0017	0.0908	0.4947	0.6233	n/a	0.0000
DTLZ7	0.9478	0.9900	0.0005	0.0005	0.5290	n/a	0.0000
ZDT3	0.0485	0.6030	0.0005	0.0005	0.0005	n/a	0.0000

G. Statistics

Table G.19.: Cogeneration study – IGD. The table shows  $p$ -values of the Anderson-Darling and Kruskal-Wallis tests. Significant results at a 5% level for the Kruskal-Wallis test are highlighted in green. Significant performance differences are observed on all problem instances.

	Anderson-Darling							Kruskal-Wallis	
	ESPEA	IBEA	MOEAD	NSGA-II	NSGA-III	SMPSO	SMS-EMOA	SPEA2	
CG0	0.0005	0.0005	0.0005	0.0005	0.0005	0.0034	0.0005	0.0005	0.0000
CG1	0.0005	0.0005	0.0005	0.0005	0.0060	0.0234	0.0005	0.0005	0.0000
CG2	0.0005	0.0005	0.0005	0.0399	0.1293	0.0016	0.0005	0.0005	0.0000
CG3	0.0005	0.0005	0.0005	0.0637	0.2012	0.0005	0.0005	0.0005	0.0000
CG4	0.0005	0.0005	0.0005	0.0077	0.0011	0.0005	0.0005	0.0005	0.0000
CG5	0.0005	0.0032	0.0005	0.0079	0.0749	0.0808	0.0005	0.0005	0.0000
CG6	0.0005	0.0005	0.0005	0.0021	0.0072	0.0024	0.0005	0.0005	0.0000
CG7	0.0174	0.0005	0.0005	0.2900	0.0435	0.0014	0.0005	0.0005	0.0000
CG8	0.0005	0.0005	0.0005	0.0371	0.0005	0.0005	0.0005	0.0005	0.0000
CG9	0.5710	0.0005	0.0005	0.0005	0.0005	0.0005	0.0005	0.0005	0.0000
CG10	0.2269	0.0005	0.0005	0.0366	0.5427	0.0012	0.0005	0.0005	0.0000
CG11	0.0005	0.0005	0.0005	0.7094	0.0028	0.0021	0.0005	0.0005	0.0000
CG12	0.0005	0.0005	0.0005	0.4539	0.0008	0.0115	0.0005	0.0005	0.0000
CG13	0.0005	0.0005	0.0005	0.0005	0.0210	0.0008	0.0005	0.0005	0.0000
CG14	0.1018	0.0005	0.0005	0.0005	0.0005	0.0005	0.0005	0.0005	0.0000
CG15	0.0098	0.0005	0.0005	0.0005	0.2587	0.0509	0.0005	0.0005	0.0000
CG16	0.0470	0.0005	0.0005	0.0005	0.0044	0.0005	0.0005	0.0005	0.0000
CG17	0.0149	0.0005	0.0005	0.0012	0.0550	0.0008	0.0005	0.0005	0.0000
CG18	0.0032	0.0005	0.0005	0.0729	0.0005	0.0005	0.0005	0.0005	0.0000
CG19	0.0005	0.0092	0.0005	0.0102	0.0024	0.0045	0.0005	0.0005	0.0000
CG20	0.0098	0.0005	0.0005	0.0026	0.0162	0.3997	0.0005	0.0005	0.0000
CG21	0.0631	0.0005	0.0005	0.0005	0.0072	0.0099	0.0005	0.0005	0.0000
CG22	0.0005	0.0005	0.0005	0.0090	0.0393	0.0005	0.0005	0.0005	0.0000
CG23	0.0005	0.0005	0.0005	0.0006	0.0022	0.0005	0.0005	0.0005	0.0000
CG24	0.0005	0.0005	0.0005	0.0005	0.0668	0.0109	0.0005	0.0005	0.0000
CG25	0.0525	0.0005	0.0005	0.0298	0.0866	0.0561	0.0005	0.0005	0.0000
CG26	0.0558	0.0005	0.0005	0.0057	0.0120	0.0006	0.0005	0.0005	0.0000
CG27	0.0005	0.0005	0.0005	0.0030	0.0635	0.0005	0.0005	0.0005	0.0000
CG28	0.0005	0.0005	0.0005	0.0005	0.0035	0.0159	0.0005	0.0005	0.0000
CG29	0.0374	0.0005	0.0005	0.3678	0.0569	0.0062	0.0031	0.0005	0.0000
CG30	0.2192	0.0005	0.0005	0.0008	0.3649	0.0005	0.0005	0.0005	0.0000
CG31	0.0228	0.0005	0.0005	0.4628	0.1153	0.0005	0.0005	0.0005	0.0000
CG32	0.1135	0.0005	0.0005	0.4391	0.0078	0.0355	0.0005	0.0005	0.0000
CG33	0.0005	0.0005	0.0005	0.0005	0.0029	0.0026	0.0005	0.0005	0.0000
CG34	0.0005	0.0005	0.0005	0.0672	0.0815	0.0260	0.0005	0.0005	0.0000
CG35	0.1272	0.0005	0.0005	0.0005	0.0262	0.0926	0.0005	0.0005	0.0000
CG36	0.0007	0.0005	0.0005	0.0005	0.0007	0.0005	0.0005	0.0005	0.0000
CG37	0.0005	0.0005	0.0005	0.0171	0.0005	0.2030	0.0005	0.0005	0.0000
CG38	0.0018	0.0005	0.0005	0.0005	0.0069	0.1136	0.0005	0.0005	0.0000

## **G.3. Interproblem Comparison**

G. Statistics

Table G.20.: Cogeneration study – IGD. The table shows  $p$ -values of the post-hoc analysis. Green cell color indicates that ESPEA outperforms the corresponding algorithm with confidence at a 95 % level. All performance differences are found to be significant.

	IBEA	MOEAD	NSGA-II	NSGA-III	SMPSO	SMS-EMOA	SPEA2
CG0	0.0188	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
CG1	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0347
CG2	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0402
CG3	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0363
CG4	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0388
CG5	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0413
CG6	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0292
CG7	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0366
CG8	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0371
CG9	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0367
CG10	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0382
CG11	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0467
CG12	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0299
CG13	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0342
CG14	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0273
CG15	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0331
CG16	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0356
CG17	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0322
CG18	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0458
CG19	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0319
CG20	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0415
CG21	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0373
CG22	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0329
CG23	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0356
CG24	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0373
CG25	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0474
CG26	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0455
CG27	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0435
CG28	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0254
CG29	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0312
CG30	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0420
CG31	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0343
CG32	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0319
CG33	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0302
CG34	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0289
CG35	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0363
CG36	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0449
CG37	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0342
CG38	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0282

## **G.4. Performance Rank Tables**

G. Statistics

Table G.21.: Performance rank table for the Friedman test in Section 5.3.2. Each column features the number of times the respective algorithm significantly outperforms another algorithm on the given test problem.

	ESPEA	IBEA	MOEAD	NSGA-II	NSGA-III	SMPSO	SMS-EMOA	SPEA2
B1	6	4	3	0	4	0	0	6
B2	5	3	3	0	5	0	2	6
B3	6	0	1	1	5	1	4	6
B4	4	5	2	0	4	0	2	6
B5	5	0	1	3	5	3	1	7
B6	5	0	2	2	5	2	0	7
DEB2DK $k = 1$	5	0	1	1	4	7	3	3
DEB2DK $k = 3$	4	0	1	1	4	7	3	4
DEB3DK $k = 1$	4	0	1	1	5	3	5	6
DEB3DK $k = 2$	5	0	1	2	6	3	2	6
DO2DK $k = 2$ $s = 1$	6	0	0	3	1	6	4	4
DO2DK $k = 4$ $s = 1$	5	0	1	3	1	6	3	4
DTLZ1	4	0	2	1	5	1	7	5
DTLZ3	6	0	1	1	6	2	1	2
DTLZ5	5	0	1	3	1	7	3	5
DTLZ7	1	0	2	4	5	3	1	7
invDTLZ1	6	0	2	4	4	3	0	6
Lamé $m = 2$ $\gamma = 0.25$	6	0	1	3	1	5	3	4
Lamé $m = 3$ $\gamma = 0.5$	6	0	1	3	0	5	3	6
Lamé $m = 2$ $\gamma = 4$	5	0	2	2	2	7	0	5
ZDT1	4	2	0	1	1	6	6	4
ZDT2	6	0	2	1	4	6	1	4
ZDT3	5	0	0	2	1	7	5	3
ZDT6	7	0	2	0	6	2	2	2

Table G.22.: Performance rank table for the Friedman test in Section 6.3.2. Each column features the number of times the respective algorithm significantly outperforms another algorithm on the given test problem.

	Sum of objectives					Nash				
	Baseline	CMA-ES	GA	HC	PSA	CMA-ES	GA	HC	PSA	
DEB2DK $k = 1$	0	2	1	3	4	0	2	1	3	4
DEB2DK $k = 2$	0	2	0	2	4	0	2	0	3	3
DEB2DK $k = 3$	2	3	0	0	4	0	2	1	3	3
DEB2DK $k = 4$	2	3	0	0	4	1	4	2	1	0
DEB3DK $k = 1$	1	0	1	3	3	0	1	2	2	2
DEB2DK $k = 2$	1	0	2	3	3	1	0	2	2	2
DO2DK $k = 1 s = 1$	0	2	1	3	4	0	2	1	3	3
DO2DK $k = 2 s = 1$	1	2	2	4	0	0	2	0	3	3
DO2DK $k = 3 s = 1$	0	2	0	3	4	0	2	0	3	3
DO2DK $k = 4 s = 1$	0	2	1	0	0	0	2	1	3	4
DTLZ7	0	0	2	3	3	0	0	2	2	2
ZDT3	0	2	1	3	4	4	1	0	1	1

	Angle utility					Tradeoff utility				
	Baseline	CMA-ES	GA	HC	PSA	CMA-ES	GA	HC	PSA	
DEB2DK $k = 1$	0	3	1	3	2	0	1	1	3	3
DEB2DK $k = 2$	0	2	1	3	3	0	2	1	3	3
DEB2DK $k = 3$	0	2	1	3	3	0	1	1	3	3
DEB2DK $k = 4$	0	2	1	3	3	0	2	2	0	4
DEB3DK $k = 1$	0	0	2	3	3	0	1	2	2	2
DEB2DK $k = 2$	1	0	2	2	2	0	0	1	1	0
DO2DK $k = 1 s = 1$	0	3	1	3	2	0	2	1	3	3
DO2DK $k = 2 s = 1$	0	3	1	3	2	0	1	1	3	3
DO2DK $k = 3 s = 1$	0	2	1	4	2	1	1	4	0	1
DO2DK $k = 4 s = 1$	0	3	1	2	3	0	1	1	3	3
DTLZ7	4	0	3	0	2	3	3	0	0	2
ZDT3	0	2	1	3	2	1	3	0	1	3



# Bibliography

- [AD52] Theodore W Anderson and Donald A Darling. Asymptotic theory of certain "goodness of fit" criteria based on stochastic processes. *The annals of mathematical statistics*, pages 193–212, 1952.
- [AD94] Ram Bhusan Agrawal and Kalyanmoy Deb. Simulated binary crossover for continuous search space. *Complex Systems*, 9(3):1–15, 1994.
- [All14] Florian Allering. *Organic Smart Home - Energiemanagement für Intelligente Gebäude*. PhD thesis, Karlsruhe Institute of Technology, 2014.
- [AP96] Timothy Ward Athan and Panos Y Papalambros. A note on weighted criteria methods for compromise solutions in multi-objective optimization. *Engineering Optimization*, 27(2):155–176, 1996.
- [APSS12] Florian Allering, Marc Premm, Pradyumn Kumar Shukla, and Hartmut Schmeck. Electrical load management in smart homes using evolutionary algorithms. In *Evolutionary Computation in Combinatorial Optimization*, pages 99–110. Springer, 2012.
- [Arr51] Kenneth J. Arrow. *Social Choice and Individual Values*. John, 1951.
- [ASO15] Hamzeh J Aljawawdeh, Christopher L Simons, and Mohammed Odeh. Metaheuristic design pattern: Preference. In *Proceedings of the Companion Publication of the 2015 Annual Conference on Genetic and Evolutionary Computation*, pages 1257–1260. ACM, 2015.
- [ASS02] Kenneth Arrow, Amartya Sen, and Kotaru Suzumura, editors. *Handbook of Social Choice and Welfare*. North-Holland, 2002.
- [Bar37] Maurice S Bartlett. Properties of sufficiency and statistical tests. *Proceedings of the royal society of London. Series A, mathematical and physical sciences*, pages 268–282, 1937.

## Bibliography

- [BBTZ13] Dimo Brockhoff, Johannes Bader, Lothar Thiele, and Eckart Zitzler. Directed multiobjective optimization based on the weighted hypervolume indicator. *Journal of Multi-Criteria Decision Analysis*, 20(5-6):291–317, 2013.
- [BD05] Jürgen Branke and Kalyanmoy Deb. Integrating user preferences into evolutionary multi-objective optimization. In *Knowledge incorporation in evolutionary computation*, pages 461–477. Springer, 2005.
- [BDDO04] Jürgen Branke, Kalyanmoy Deb, Henning Dierolf, and Matthias Osswald. Finding knees in multi-objective optimization. In *Parallel Problem Solving from Nature - PPSN VIII*, pages 722–731. Springer, 2004.
- [BDMS08] Jürgen Branke, Kalyanmoy Deb, Kaisa Miettinen, and Roman Slowinski, editors. *Multiobjective optimization: Interactive and evolutionary approaches*. Springer, 2008.
- [BDMS16] Marlon Alexander Braun, Thomas Dengiz, Ingo Mauser, and Hartmut Schmeck. Comparison of multi-objective evolutionary optimization in smart building scenarios. In *European Conference on the Applications of Evolutionary Computation*, pages 443–458. Springer, Cham, 2016.
- [Bel86] David E. Bell. Double-exponential utility functions. *Mathematics of Operations Research*, 11(2):351–361, 1986.
- [Ber99] Dimitri Bertsekas. *Nonlinear programming*. Athena scientific Belmont, 1999.
- [BF03] Ş İlker Birbil and Shu-Chering Fang. An electromagnetism-like mechanism for global optimization. *Journal of global optimization*, 25(3):263–282, 2003.
- [BF09] Karl Bringmann and Tobias Friedrich. Approximating the least hypervolume contributor: NP-hard in general, but fast in practice. In *Evolutionary Multi-Criterion Optimization*, pages 6–20. Springer, 2009.
- [BHSS17] Marlon Alexander Braun, Lars Heling, Pradyumn Shukla, and Hartmut Schmeck. Multimodal scalarized preferences in multi-objective optimization. In *Proceedings of the 2017 on Genetic and Evolutionary Computation Conference*. ACM, 2017.
- [BKS01] Jürgen Branke, Thomas Kaußler, and Hartmut Schmeck. Guidance in evolutionary multi-objective optimization. *Advances in Engineering Software*, 32(6):499–507, 2001.

- [BKSG15] Slim Bechikh, Marouane Kessentini, Lamjed Ben Said, and Khaled Ghédira. Chapter four-preference incorporation in evolutionary multiobjective optimization: A survey of the state-of-the-art. *Advances in Computers*, 98:141–207, 2015.
- [BNE07] Nicola Beume, Boris Naujoks, and Michael Emmerich. SMS-EMOA: Multiobjective selection based on dominated hypervolume. *European Journal of Operational Research*, 181(3):1653–1669, 2007.
- [Bou54] Nicolas Bourbaki. *Éléments de mathématique: Les structures fondamentales de l'analyse. Théorie des ensembles*. Hermann, 1954.
- [Bra] Marlon Alexander Braun. jMetal Plus. <https://sourceforge.net/projects/jmetalbymarlonso/>. Accessed 29.12.2017.
- [Bra14] Marlon A. Braun. Theory and algorithms for identifying knees in multi-objective optimization. Master's thesis, Karlsruhe Institute of Technology, 2014.
- [Bri22] Percy Williams Bridgman. *Dimensional analysis*. Yale University Press, 1922.
- [Bri81] Anne Brindle. *Genetic algorithms for function optimization*. PhD thesis, Edmonton: University of Alberta, Department of Computer Science, 1981.
- [BS07] Hans-Georg Beyer and Bernhard Sendhoff. Robust optimization – a comprehensive survey. *Computer Methods in Applied Mechanics and Engineering*, 196(33):3190 – 3218, 2007.
- [BS16] Marlon Alexander Braun and Pradyumn Kumar Shukla. On cone based decompositions of proper Pareto optimality. *Optimization Online*, 2016.
- [BSE<sup>+</sup>16] Marlon Alexander Braun, Sandra Seijo, Javier Echanobe, Pradyumn Kumar Shukla, Ines del Campo, Javier Garcia-Sedano, and Harmut Schmeck. A neuro-genetic approach for modeling and optimizing a complex cogeneration process. *Applied Soft Computing*, 48:347 – 358, 2016.
- [BSS11] Marlon Alexander Braun, Pradyumn Kumar Shukla, and Hartmut Schmeck. Preference ranking schemes in multi-objective evolutionary algorithms. In Ricardo H.C. Takahashi, Kalyanmoy Deb, Elizabeth F. Wanner, and Salvatore Greco, editors, *Evolutionary Multi-Criterion Optimization*, volume 6576 of LNCS, pages 226–240. Springer, 2011.

## Bibliography

- [BSS15] Marlon Alexander Braun, Pradyumn Kumar Shukla, and Hartmut Schmeck. Obtaining optimal pareto front approximations using scalarized preference information. In *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation, GECCO '15*, pages 631–638, New York, NY, USA, 2015. ACM.
- [BSS17] Marlon Alexander Braun, Pradyumn Shukla, and Hartmut Schmeck. Angle-based preference models in multi-objective optimization. In *International Conference on Evolutionary Multi-Criterion Optimization*, pages 88–102. Springer, 2017.
- [BZ11] Johannes Bader and Eckart Zitzler. Hype: An algorithm for fast hypervolume-based many-objective optimization. *Evolutionary Computation*, 19(1):45–76, 2011.
- [CAA14] Pedro Carrasqueira, Maria João Alves, and Carlos Henggeler Antunes. An improved multiobjective electromagnetism-like mechanism algorithm. In *Applications of Evolutionary Computation*, pages 627–638. Springer, 2014.
- [CAA15] Pedro Carrasqueira, Maria João Alves, and Carlos Henggeler Antunes. A multiobjective electromagnetism-like algorithm with improved local search. In *Operational Research*, pages 123–144. Springer, 2015.
- [Can82] Georg Cantor. Über unendliche, lineare punktmannichfaltigkeiten. *Mathematische Annalen*, 20(1):113–121, 1882.
- [CCLVV07] Carlos Coello Coello, Gary Lamont, and David Van Veldhuizen. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Springer, 2007.
- [CD28] Charles W Cobb and Paul H Douglas. A theory of production. *The American Economic Review*, 18(1):139–165, 1928.
- [Cho54] Gustave Choquet. Theory of capacities. In *Annales de l'institut Fourier*, volume 5, pages 131–295. Association des Annales de l'Institut Fourier, 1954.
- [CM75] Jared L Cohon and David H Marks. A review and evaluation of multiobjective programming techniques. *Water Resources Research*, 11(2):208–220, 1975.
- [cma] [https://www.lri.fr/~hansen/cmaes\\_inmatlab.html#practical](https://www.lri.fr/~hansen/cmaes_inmatlab.html#practical). Accessed 23.10.2017.

- [Coe00] CA Coello Coello. Handling preferences in evolutionary multi-objective optimization: A survey. In *Evolutionary Computation, 2000. Proceedings of the 2000 Congress on*, volume 1, pages 30–37. IEEE, 2000.
- [Coe02] Carlos A Coello Coello. Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art. *Computer methods in applied mechanics and engineering*, 191(11):1245–1287, 2002.
- [Cor09] Thomas H Cormen. *Introduction to algorithms*. MIT press, 2009.
- [CSB<sup>+</sup>12] Michael P Cipold, Pradyumn Kumar Shukla, Claus C Bachmann, Kaibin Bao, and Hartmut Schmeck. An evolutionary optimization approach for bulk material blending systems. In *Parallel Problem Solving from Nature-PPSN XII*, pages 478–488. Springer, 2012.
- [Dal13] David Dallinger. The contribution of vehicle-to-grid to balance fluctuating generation: Comparing different battery ageing approaches. Technical report, Working Paper Sustainability and Innovation, 2013.
- [Das99] Indraneel Das. On characterizing the "knee" of the pareto curve based on normal-boundary intersection. *Structural Optimization*, 18(2-3):107–115, 1999.
- [DD97] I. Das and J.E. Dennis. A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multicriteria optimization problems. *Structural optimization*, 14(1):63–69, 1997.
- [DD98] Indraneel Das and John E Dennis. Normal-boundary intersection: A new method for generating the pareto surface in nonlinear multicriteria optimization problems. *SIAM Journal on Optimization*, 8(3):631–657, 1998.
- [Deb01] Kalyanmoy Deb. *Multi-Objective optimization using evolutionary algorithms*, volume 16. John Wiley & Sons, 2001.
- [Deb03] Kalyanmoy Deb. *Multi-objective Evolutionary Algorithms: Introducing Bias Among Pareto-optimal Solutions*, pages 263–292. Springer Berlin Heidelberg, Berlin, Heidelberg, 2003.
- [DG96] Kalyanmoy Deb and Mayank Goyal. A combined genetic adaptive search (GeneAS) for engineering design. *Computer Science and Informatics*, 26:30–45, 1996.

## Bibliography

- [dG02] Claude d'Aspremont and Louis Gevers. Chapter 10 social welfare functionals and interpersonal comparability. In *Handbook of Social Choice and Welfare*, volume 1 of *Handbook of Social Choice and Welfare*, pages 459 – 541. Elsevier, 2002.
- [DG11] Kalyanmoy Deb and Shivam Gupta. Understanding knee points in bicriteria problems and their implications as preferred solution principles. *Engineering optimization*, 43(11):1175–1204, 2011.
- [DJ14] Kalyanmoy Deb and Himanshu Jain. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: Solving problems with box constraints. *Evolutionary Computation, IEEE Transactions on*, 18(4):577–601, 2014.
- [DM60] Augustus De Morgan. *Syllabus of a proposed system of logic*. Walton and Maberly, 1860.
- [DN11] Juan J. Durillo and Antonio J. Nebro. jMetal: A java framework for multi-objective optimization. *Advances in Engineering Software*, 42:760–771, 2011.
- [DNLA09] Juan José Durillo, Antonio J. Nebro, Francisco Luna, and Enrique Alba. On the effect of the steady-state selection scheme in multi-objective genetic algorithms. In *EMO*, pages 183–197. Springer, 2009.
- [DPAM02] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *Evolutionary Computation, IEEE Transactions on*, 6(2):182–197, 2002.
- [DS05] Kalyanmoy Deb and Dhish Kumar Saxena. On finding pareto-optimal solutions through dimensionality reduction for certain large-dimensional multi-objective optimization problems. Technical report, Kanpur Genetic Algorithms Laboratory, 2005.
- [DTLZ05] Kalyanmoy Deb, Lothar Thiele, Marco Laumanns, and Eckart Zitzler. Scalable test problems for evolutionary multiobjective optimization. In Lakhmi Jain, Xindong Wu, Ajith Abraham, Lakhmi Jain, and Robert Goldberg, editors, *Evolutionary Multiobjective Optimization*, Advanced Information and Knowledge Processing, pages 105–145. Springer, 2005.
- [Dun64] Olive Jean Dunn. Multiple comparisons using rank sums. *Technometrics*, 6(3):241–252, 1964.

- [ED07] Michael TM Emmerich and André H Deutz. Test problems based on Lamé superspheres. In *Evolutionary Multi-Criterion Optimization*, pages 922–936. Springer, 2007.
- [Edg81] Francis Ysidro Edgeworth. *Mathematical psychics: An essay on the application of mathematics to the moral sciences*, volume 10. Kegan Paul, 1881.
- [EH89] Jon Elster and Aanund Hylland. *Foundations of social choice theory*. CUP Archive, 1989.
- [Ehr05] Matthias Ehrgott. *Multicriteria optimization*, volume 2. Springer, 2005.
- [Eic12] Gabriele Eichfelder. *Variable ordering structures in vector optimization*. Springer, 2012.
- [Eic14] Gabriele Eichfelder. Numerical procedures in multiobjective optimization with variable ordering structures. *Journal of Optimization Theory and Applications*, 162(2):489–514, 2014.
- [Ert11] Wolfgang Ertel. *Introduction to artificial intelligence*. Springer Science & Business Media, 2011.
- [ES<sup>+</sup>03] Agoston E Eiben, James E Smith, et al. *Introduction to evolutionary computing*, volume 53. Springer, 2003.
- [Fau] Kjell Magne Fauske. Neural network illustration. <http://www.texample.net/tikz/examples/neural-network/>. Accessed: 06.11.2017.
- [FF96] Carlos M Fonseca and Peter J Fleming. On the performance assessment and comparison of stochastic multiobjective optimizers. In *Parallel problem solving from nature - PPSN IV*, pages 584–593. Springer, 1996.
- [FKN13] Tobias Friedrich, Trent Kroeger, and Frank Neumann. Weighted preferences in evolutionary multi-objective optimization. *International Journal of Machine Learning and Cybernetics*, 4(2):139–148, Apr 2013.
- [FKPT12] Ludwig Fahrmeier, Rita Künstler, Iris Pigeot, and Gerhard Tutz. *Statistik: Der Weg zu Datenanalyse*. Springer, 2012.
- [Flu13] Dandolo Flumini. *Weak well orders*. PhD thesis, Universität Bern, 2013.

## Bibliography

- [Fri37] Milton Friedman. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the American Statistical Association*, 32(200):675–701, 1937.
- [GC17] Raquel Hernández Gómez and Carlos A. Coello Coello. A hyper-heuristic of scalarizing functions. In *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO '17*, pages 577–584, New York, NY, USA, 2017. ACM.
- [GEF16] Salvatore Greco, Matthias Ehrgott, and José Figueira, editors. *Multiple criteria decision analysis: state of the art surveys*. Springer Science & Business Media, 2016.
- [Geo68] A. M. Geoffrion. Proper efficiency and the theory of vector maximization. *Journal of Mathematical Analysis and Applications*, 22:618–630, 1968.
- [Gra96] Michel Grabisch. The application of fuzzy integrals in multicriteria decision making. *European Journal of Operational Research*, 89(3):445 – 456, 1996.
- [Han06] Nikolaus Hansen. The cma evolution strategy: A comparing review. In Jose A. Lozano, Pedro Larrañaga, Iñaki Inza, and Endika Bengoetxea, editors, *Towards a New Evolutionary Computation: Advances in the Estimation of Distribution Algorithms*, pages 75–102. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.
- [HHBW06] Simon Huband, Philip Hingston, Luigi Barone, and Lyndon While. A review of multiobjective test problems and a scalable test problem toolkit. *IEEE Transactions on Evolutionary Computation*, 10(5):477–506, 2006.
- [Hil01] Claus Hillermeier. *Nonlinear multiobjective optimization*, volume 135 of *International Series of Numerical Mathematics*. Birkhäuser Verlag, Basel, 2001.
- [HJL15] Lixia Han, Shujuan Jiang, and Shaojiang Lan. Novel electromagnetism-like mechanism method for multiobjective optimization problems. *Journal of Systems Engineering and Electronics*, 26(1):182–189, 2015.
- [HR88] Terry P Harrison and Richard E Rosenthal. Optimizability of utility and value functions. *Naval Research Logistics (NRL)*, 35(3):411–418, 1988.
- [HRW10] David Halliday, Robert Resnick, and Jearl Walker. *Fundamentals of physics extended*. John Wiley & Sons, 2010.

- [IHK12] GH Isazadeh, R Hooshmand, and A Khodabakhshian. Design of an adaptive dynamic load shedding algorithm using neural network in the steelmaking cogeneration facility. *Iranian Journal of Science and Technology, Transactions of Electrical Engineering*, 36(1):67–82, 2012.
- [IHR07] Christian Igel, Nikolaus Hansen, and Stefan Roth. Covariance matrix adaptation for multi-objective optimization. *Evolutionary computation*, 15(1):1–28, 2007.
- [IISN17] Hisao Ishibuchi, Ryo Imada, Yu Setoguchi, and Yusuke Nojima. Hypervolume subset selection for triangular and inverted triangular pareto fronts of three-objective problems. In *Proceedings of the 14th ACM/SIGEVO Conference on Foundations of Genetic Algorithms*, pages 95–110. ACM, 2017.
- [ITN08] Hisao Ishibuchi, Noritaka Tsukamoto, and Yusuke Nojima. Evolutionary many-objective optimization: A short review. In *IEEE congress on evolutionary computation*, pages 2419–2426. Citeseer, 2008.
- [JD14] Himanshu Jain and Kalyanmoy Deb. An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part II: Handling constraints and extending to an adaptive approach. *Evolutionary Computation, IEEE Transactions on*, 18(4):602–622, 2014.
- [JLS82] E. Jacquet-Lagrange and J. Siskos. Assessing a set of additive utility functions for multicriteria decision-making, the uta method. *European Journal of Operational Research*, 10(2):151 – 164, 1982.
- [Kar72] Richard M Karp. Reducibility among combinatorial problems. In *Complexity of computer computations*, pages 85–103. Springer, 1972.
- [KD06] Saku Kukkonen and Kalyanmoy Deb. Improved pruning of non-dominated solutions based on crowding distance for bi-objective optimization problems. In *Proceedings of the World Congress on Computational Intelligence*, pages 1179–1186, Vancouver, Canada, 2006. IEEE Press.
- [Kee09] Ralph L Keeney. *Value-focused thinking: A path to creative decisionmaking*. Harvard University Press, 2009.
- [Kha14] Majid Khalili. A multi-objective electromagnetism algorithm for a bi-objective hybrid no-wait flowshop scheduling problem.

## Bibliography

- The International Journal of Advanced Manufacturing Technology*, 70(9):1591–1601, Feb 2014.
- [Kno06] J. Knowles. Parego: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation*, 10(1):50–66, Feb 2006.
- [KOW04] Michael M Kostreva, Włodzimierz Ogryczak, and Adam Wierzbicki. Equitable aggregations and multiple criteria analysis. *European Journal of Operational Research*, 158(2):362–377, 2004.
- [KR93] Ralph L Keeney and Howard Raiffa. *Decisions with multiple objectives: preferences and value trade-offs*. Cambridge university press, 1993.
- [KT79] Daniel Kahneman and Amos Tversky. Prospect theory: An analysis of decision under risk. *Econometrica: Journal of the Econometric Society*, pages 263–291, 1979.
- [KW52] William H Kruskal and W Allen Wallis. Use of ranks in one-criterion variance analysis. *Journal of the American statistical Association*, 47(260):583–621, 1952.
- [Lee10] John Lee. *Introduction to topological manifolds*, volume 940. Springer Science & Business Media, 2010.
- [LFWB03] Yusheng Li, Georges M. Fadel, Margaret Wiecek, and Vincent Y. Blouin. Minimum effort approximation of the pareto space of convex bi-criteria problems. *Optimization and Engineering*, 4(3):231–261, Sep 2003.
- [LYBF<sup>+</sup>16] Longmei Li, Iryna Yevseyeva, Vitor Basto-Fernandes, Heike Trautmann, Ning Jing, and Michael Emmerich. An ontology of preference-based multiobjective evolutionary algorithms. *arXiv preprint arXiv:1609.08082*, 2016.
- [LYBF<sup>+</sup>17] Longmei Li, Iryna Yevseyeva, Vitor Basto-Fernandes, Heike Trautmann, Ning Jing, and Michael Emmerich. Building and using an ontology of preference-based multiobjective evolutionary algorithms. In Heike Trautmann, Günter Rudolph, Kathrin Klamroth, Oliver Schütze, Margaret Wiecek, Yaochu Jin, and Christian Grimme, editors, *Evolutionary Multi-Criterion Optimization: 9th International Conference, EMO 2017, Münster, Germany, March 19-22, 2017, Proceedings*, pages 406–421. Springer International Publishing, Cham, 2017.

- [LZ09] Hui Li and Qingfu Zhang. Multiobjective optimization problems with complicated pareto sets, MOEA/D and NSGA-II. *Evolutionary Computation, IEEE Transactions on*, 13(2):284–302, 2009.
- [MA04] R Timothy Marler and Jasbir S Arora. Survey of multi-objective optimization methods for engineering. *Structural and multidisciplinary optimization*, 26(6):369–395, 2004.
- [MBC79] M. D. McKay, R. J. Beckman, and W. J. Conover. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21(2):239–245, 1979.
- [Mes96] Achille Messac. Physical programming-effective optimization for computational design. *AIAA journal*, 34(1):149–158, 1996.
- [Mie99] Kaisa Miettinen. *Nonlinear multiobjective optimization*, volume 12. Springer, 1999.
- [MMAS16] Ingo Mauser, Jan Müller, Florian Allering, and Hartmut Schmeck. Adaptive building energy management with multiple commodities and flexible evolutionary optimization. *Renewable Energy*, 87(Part 2):911 – 921, 2016. Optimization Methods in Renewable Energy Systems Design.
- [MMD91] Ravi Mazumdar, Lorne G Mason, and Christos Douligeris. Fairness in network optimal flow control: Optimality of product forms. *IEEE Transactions on communications*, 39(5):775–782, 1991.
- [MMVRCC06] Efrñn Mezura-Montes, Jesús Velázquez-Reyes, and Carlos A. Coello Coello. A comparative study of differential evolution variants for global optimization. In *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation, GECCO '06*, pages 485–492, New York, NY, USA, 2006. ACM.
- [MPSM00] Achille Messac, Cyriaque Puemi-Sukam, and Emanuel Melachrinoudis. Aggregate objective functions and Pareto frontiers: Required relationships and practical implications. *Optimization and Engineering*, 1(2):171–188, 2000.
- [MW47] Henry B Mann and Donald R Whitney. On a test of whether one of two random variables is stochastically larger than the other. *The annals of mathematical statistics*, pages 50–60, 1947.
- [Nas50] John Forbes Nash. The bargaining problem. *Econometrica: Journal of the Econometric Society*, pages 155–162, 1950.

## Bibliography

- [NDGN<sup>+</sup>09] A.J. Nebro, J.J. Durillo, J. Garcia-Nieto, C.A. Coello Coello, F. Luna, and E. Alba. SMPSO: A new PSO-based metaheuristic for multi-objective optimization. In *IEEE symposium on computational intelligence in multi-criteria decision-making, 2009. MCDM '09*, pages 66–73, March 2009.
- [NLA<sup>+</sup>08] A.J. Nebro, F. Luna, E. Alba, B. Dorronsoro, J.J. Durillo, and A. Beham. AbYSS: Adapting scatter search to multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 12(4):439–457, Aug 2008.
- [NSH<sup>+</sup>13] Iman Noshadi, Abdolhamid Salahi, Mahmood Hemmati, Fatemeh Rekabdar, and Toraj Mohammadi. Experimental and anfis modeling for fouling analysis of oily wastewater treatment using ultrafiltration. *Asia-Pacific Journal of Chemical Engineering*, 8(4):527–538, 2013.
- [Ogr09] Włodzimierz Ogryczak. Inequality measures and equitable locations. *Annals of Operations Research*, 167(1):61–86, 2009.
- [Pad13] Dhanesh Padmanabhan. A clustering-based methodology for discontinuous pareto frontier representation. *Journal of Multi-Criteria Decision Analysis*, 20(5-6):235–253, 2013.
- [Par96] Vilfredo Pareto. *Cours d'économie politique*. Librairie Droz, 1896.
- [PGE09] Anthony Przybylski, Xavier Gandibleux, and Matthias Ehrgott. Computational results for four exact methods to solve the three-objective assignment problem. *Lect. Notes Econ. Math. Syst.*, 618:79–88, 2009.
- [PKB07] Riccardo Poli, James Kennedy, and Tim Blackwell. Particle swarm optimization. *Swarm intelligence*, 1(1):33–57, 2007.
- [Pre15] Mike Preuss. *Multimodal Optimization by Means of Evolutionary Algorithms*. Springer, 2015.
- [Raw71] John Rawls. *A Theory of Justice*. Harvard, 1971.
- [RM06] Laura C Rodman and Ross K Meentemeyer. A geographic analysis of wind turbine placement in northern california. *Energy Policy*, 34(15):2137–2149, 2006.
- [Roe98] John E Roemer. *Theories of distributive justice*. Harvard University Press, 1998.
- [Roy96] Bernard Roy. *Multicriteria Methodology for Decision Aiding*. Springer, 1996.

- [RS06] L. Rachmawati and D. Srinivasan. Preference incorporation in multi-objective evolutionary algorithms: A survey. In *2006 IEEE International Conference on Evolutionary Computation*, pages 962–968, 2006.
- [RVMB14] Francesco Rossi, David Velázquez, Iñigo Monedero, and Félix Biscarri. Artificial neural networks and physical modeling for determination of baseline consumption of CHP plants. *Expert Systems with Applications*, 41(10):4658 – 4669, 2014.
- [RW05] S. Ruzika and M. M. Wiecek. Approximation methods in multiobjective programming. *Journal of Optimization Theory and Applications*, 126(3):473–501, Sep 2005.
- [SB13] Pradyumn Kumar Shukla and Marlon Alexander Braun. Indicator based search in variable orderings: Theory and algorithms. In Robin C. Purshouse, Peter J. Fleming, Carlos M. Fonseca, Salvatore Greco, and Jane Shaw, editors, *Evolutionary Multi-Criterion Optimization*, volume 7811 of *Lecture Notes in Computer Science*, pages 66–80. Springer, 2013.
- [SBS13] Pradyumn Kumar Shukla, Marlon Alexander Braun, and Hartmut Schmeck. Theory and algorithms for finding knees. In Robin C. Purshouse, Peter J. Fleming, Carlos M. Fonseca, Salvatore Greco, and Jane Shaw, editors, *Evolutionary Multi-Criterion Optimization*, volume 7811 of *LNCS*, pages 156–170. Springer, 2013.
- [SBS14] Pradyumn Kumar Shukla, Marlon Alexander Braun, and Hartmut Schmeck. On the interrelationships between knees and aggregate objective functions. In *Proceedings of the 2014 Conference Companion on Genetic and Evolutionary Computation Companion*, GECCO Comp '14, pages 95–96, New York, NY, USA, 2014. ACM.
- [SCBS14] Pradyumn Kumar Shukla, Michael P Cipold, Claus Bachmann, and Hartmut Schmeck. On homogenization of coal in longitudinal blending beds. In *Proceedings of the 2014 conference on Genetic and evolutionary computation*, pages 1199–1206. ACM, 2014.
- [Sch13] Richard Evan Schwartz. The five-electron case of thomson’s problem. *Experimental Mathematics*, 22(2):157–186, 2013.
- [SDS14] Pradyumn Kumar Shukla, Nadja Doll, and Hartmut Schmeck. A theoretical analysis of volume based pareto front approximations. In *Proceedings of the 2014 conference on Genetic and evolutionary computation*, pages 1415–1422. ACM, 2014.

## Bibliography

- [SED13] Pradyumn Kumar Shukla, Michael Emmerich, and André Deutz. A theoretical analysis of curvature based preference models. In Robin C. Purshouse, Peter J. Fleming, CarlosM. Fonseca, Salvatore Greco, and Jane Shaw, editors, *Evolutionary Multi-Criterion Optimization*, volume 7811 of *Lecture Notes in Computer Science*, pages 367–382. Springer, 2013.
- [Sen70] A Sen. Collective choice and social welfare holden-day san francisco. *California Google Scholar*, 1970.
- [SHS10a] Pradyumn Kumar Shukla, Christian Hirsch, and Hartmut Schmeck. A framework for incorporating trade-off information using multi-objective evolutionary algorithms. In *Parallel Problem Solving from Nature, PPSN XI*, pages 131–140. Springer, 2010.
- [SHS10b] Pradyumn Kumar Shukla, Christian Hirsch, and Hartmut Schmeck. In *Search of Equitable Solutions Using Multi-objective Evolutionary Algorithms*, pages 687–696. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.
- [Shu07] Pradyumn Kumar Shukla. In search of proper pareto-optimal solutions using multi-objective evolutionary algorithms. In *Computational Science–ICCS 2007*, pages 1013–1020. Springer, 2007.
- [Sia16] Patrick Siarry, editor. *Metaheuristics*. Springer, 2016.
- [SIR11] Hemant Kumar Singh, Amitay Isaacs, and Tapabrata Ray. A pareto corner search evolutionary algorithm and dimensionality reduction in many-objective optimization problems. *IEEE Transactions on Evolutionary Computation*, 15(4):539–556, 2011.
- [SK97] E. B. Saff and A. B. J. Kuijlaars. Distributing many points on a sphere. *The Mathematical Intelligencer*, 19(1):5–11, 1997.
- [Som81] Dan M. Somers. Design and experimental results for a natural-laminar-flow airfoil for general aviation applications. Technical report, NASA, 1981.
- [Sör15] Kenneth Sörensen. Metaheuristics—the metaphor exposed. *International Transactions in Operational Research*, 22(1):3–18, 2015.
- [SP97] Rainer Storn and Kenneth Price. Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11(4):341–359, 1997.

- [SS13] Deepak Sharma and Prem Soren. Infeasibility driven approach for bi-objective evolutionary optimization. In *Evolutionary Computation (CEC), 2013 IEEE Congress on*, pages 868–875. IEEE, 2013.
- [Sta79] Wolfram Stadler. A survey of multicriteria optimization or the vector maximum problem, part I: 1776–1960. *Journal of Optimization Theory and Applications*, 29(1):1–52, 1979.
- [SW13] S. Sudeng and N. Wattanapongsakorn. Adaptive geometric angle-based algorithm with independent objective biasing for pruning pareto-optimal solutions. In *Science and Information Conference (SAI), 2013*, pages 514–523, Oct 2013.
- [SWSJ11] Kin Cheong Sou, James Weimer, Henrik Sandberg, and Karl Henrik Johansson. Scheduling smart home appliances using mixed integer linear programming. In *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*, pages 5144–5149. IEEE, 2011.
- [TF07] Tea Tušar and Bogdan Filipič. Differential evolution versus genetic algorithms in multiobjective optimization. In *Evolutionary multi-criterion optimization*, pages 257–271. Springer, 2007.
- [TFD11] Santosh Tiwari, Georges Fadel, and Kalyanmoy Deb. AMGA2: Improving the performance of the archive-based micro-genetic algorithm for multi-objective optimization. *Engineering Optimization*, 43(4):377–401, 2011.
- [Tho04] Joseph John Thomson. Xxiv. on the structure of the atom: an investigation of the stability and periods of oscillation of a number of corpuscles arranged at equal intervals around the circumference of a circle; with application of the results to the theory of atomic structure. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 7(39):237–265, 1904.
- [TK06] Ching-Shih Tsou and Chia-Hung Kao. An electromagnetism-like meta-heuristic for multi-objective optimization. In *Evolutionary Computation, 2006. CEC 2006. IEEE Congress on*, pages 1172–1178. IEEE, 2006.
- [TMKM09] Lothar Thiele, Kaisa Miettinen, Pekka J Korhonen, and Julian Molina. A preference-based evolutionary algorithm for multi-objective optimization. *Evolutionary computation*, 17(3):411–436, 2009.

## Bibliography

- [TV92] Aimo Törn and Sami Viitanen. Topographical global optimization. *Recent advances in global optimization*, pages 384–398, 1992.
- [VNM53] John Von Neumann and Oskar Morgenstern. Theory of games and economic behavior. *Princeton university press*, 1953.
- [VR10] Hal R Varian and Jack Repcheck. *Intermediate microeconomics: a modern approach*, volume 6. WW Norton & Company New York, 2010.
- [VVL98] David A. Van Veldhuizen and Gary B. Lamont. Evolutionary computation and convergence to a pareto front. In *Late breaking papers at the genetic programming 1998 conference*, pages 221–228. Citeseer, 1998.
- [WB93] Martin Weber and Katrin Borcherdig. Behavioral influences on weight judgments in multiattribute decision making. *European Journal of Operational Research*, 67(1):1–12, 1993.
- [WCL10] Upali K. Wickramasinghe, Robert Carrese, and Xiaodong Li. Designing airfoils using a reference point based evolutionary many-objective particle swarm optimization algorithm. In *Evolutionary Computation (CEC), 2010 IEEE Congress on*, pages 1–8. IEEE, 2010.
- [Wes15] Simon Wessing. *Two-stage methods for multimodal optimization*. PhD thesis, 2015.
- [Wie07] Margaret M. Wiecek. Advances in cone-based preference modeling for decision making with multiple criteria. *Decision Making in Manufacturing and Services*, 1(1-2):153–173, 2007.
- [WM97] David H Wolpert and William G Macready. No free lunch theorems for optimization. *IEEE transactions on evolutionary computation*, 1(1):67–82, 1997.
- [WRP16] Simon Wessing, Günter Rudolph, and Mike Preuss. Assessing basin identification methods for locating multiple optima. In *Advances in Stochastic and Deterministic Global Optimization*, pages 53–70. Springer, 2016.
- [YE10] Alkın Yurtkuran and Erdal Emel. A new hybrid electromagnetism-like algorithm for capacitated vehicle routing problems. *Expert Systems with Applications*, 37(4):3427 – 3433, 2010.

- [ZA12] Yuanyuan Zhang and Jiaoyan Ai. Zeta potential modeling of papermaking wastewater on neural network. In *Instrumentation, Measurement, Computer, Communication and Control (IM-CCC), 2012 Second International Conference on*, pages 63–66, Dec 2012.
- [ZDT00] Eckart Zitzler, Kalyanmoy Deb, and Lothar Thiele. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary computation*, 8(2):173–195, 2000.
- [ZK04] Eckart Zitzler and Simon Künzli. Indicator-based selection in multiobjective search. In *Parallel Problem Solving from Nature-PPSN VIII*, pages 832–842. Springer, 2004.
- [ZL07] Qingfu Zhang and Hui Li. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *Evolutionary Computation, IEEE Transactions on*, 11(6):712–731, December 2007.
- [ZLT01] Eckart Zitzler, Marco Laumanns, and Lothar Thiele. Spea2: Improving the strength pareto evolutionary algorithm. Technical Report 103, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, 2001.
- [ZT99] Eckart Zitzler and Lothar Thiele. Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach. *Evolutionary Computation, IEEE Transactions on*, 3(4):257–271, Nov 1999.