Multi-Objective Mathematical Optimization in Assisted Production Planning

Louis Schäfer (louis.schaefer@kit.edu)
wbk Institute of Production Science
Karlsruhe Institute of Technology KIT, Germany

Stefan Tse
wbk Institute of Production Science
Karlsruhe Institute of Technology KIT, Germany

Marvin Carl May
wbk Institute of Production Science
Karlsruhe Institute of Technology KIT, Germany

Gisela Lanza
wbk Institute of Production Science
Karlsruhe Institute of Technology KIT, Germany

Abstract

In today's fast-paced technological landscape, products are constantly evolving, and mass customization is providing customers with personalized goods. However, despite these advancements, production planning processes in manufacturing companies are still predominantly manual and time-consuming. The need for increased efficiency in planning becomes crucial as the frequency of production planning activities rises due to shorter time-to-market and higher product variance. Addressing the complex challenge of line balancing, the article highlights the limitations of manual planning in Excel and advocates for the application of *Operations Research* (OR) methods through a novel research approach. The proposed methodology aims to use multi-objective mathematical optimization to systematically find solutions for *Assembly Line Design* (ALD), providing a more efficient alternative to traditional manual planning with the ability to quantitatively compare various optimization criteria. This work thus provides an essential basis for the optimization of a complex closed-loop factory, as planning in remanufacturing considers uncertainties with constant reconfiguration.

Keywords: Assisted Production Planning, Operational Excellence, Mathematical Optimization, Circular Factory, Remanufacturing

Introduction & Purpose

In an era dominated by rapid technological advancements, developed products are improving and mass customization offers customer individual goods. On the other hand, technological developments in production planning remain slow. In most manufacturing companies, labor-intensive and time-consuming manual planning processes are still prevalent today. With a shorter time-to-market and higher product variance, the frequency of production planning activities increases which calls for more efficiency in planning processes. One crucial part of production system planning is line balancing, which addresses the problem of assigning process steps (i.e. tasks) to machines (i.e. stations) while optimizing one or multiple target values such as costs, quality, flexibility or else. Here, manual planning in Excel reaches its limits and the potential to solve the problem explicitly by means of mathematical optimization using *Operations Research* (OR) methods becomes apparent.

Therefore, this article proposes a novel research approach to develop, initialize, implement and apply an OR model to solve the *Assembly Line Design* (ALD) in order to assist humans in production system planning. The purpose of this research is to use multi-objective mathematical optimization to systematically identify an optimal solution instead of time-consuming, manual line balancing, which ultimately does not allow any quantitative comparison of alternative solutions with regards to multiple optimization criteria. The developed research methodology is applied to a real-world example from the industry and the research design is described in the following section.

Methodological Approach

The chosen methodological approach (Fig. 1) provides for a combination of analysis, description and prescription and is based on the phases of the *Design Research Methodology* (DRM) according to BLESSING AND CHAKRABARTI (2009). After analyzing the relevant fields of research in a literature study and deriving the research questions (I), the problem is defined (II) in order to develop concepts and to implement (III) and evaluate (IV) them. How the methodology was adapted and applied is described below.

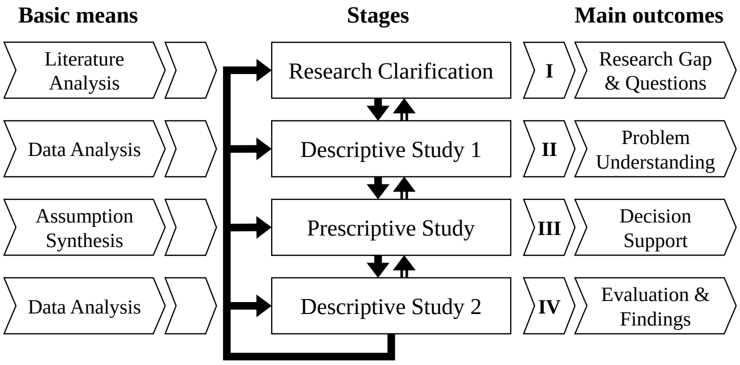

Figure 1 – Methodology

I. Fields of Action

Firstly, on the basis of a comprehensive literature analysis, the object of research and the current state of the art in resource allocation problems were identified.

Resource allocation is an essential part of production system planning. Methods of OR can assist planners in finding an optimal resource allocation and consider explicitly additional aspects (e.g. flexibility, energy) beyond the conventional approach. Due to their complexity, resource allocation problems are classified as NP-hard by ÁLVAREZ-MIRANDA & PEREIRA (2019), and as such, a solution can not be generated in polynomial time. Therefore, various solution algorithms exist, which can be categorized into optimal algorithms and heuristics.

The following provides an explanation of *Job Shop Scheduling* (JSS), *Assembly Line Balancing* (ALB) and ALD, along with an overview of the current literature.

VAN LAARHOVEN ET AL (1992) describes JSS as the allocation of processes to several machines from jobs with different process sequences. ALKHATEEB ET AL (2022) present an algorithm that combines the optimization operators of *Cuckoo Search algorithm* (CS) and the *Simulated Annealing Algorithm* (SAA) to solve JSS. Based on the original version of CS according to YANG & DEB (2009), this algorithm relies on the parasitic reproductive behaviour of cuckoo birds, where they lay their eggs in other bird´s nests. The original SAA by KIRKPATRICK ET AL (1983) models the cooling process of metals, where a balance between heating and cooling the metal is established to shape the metal into the desired shape. Another approach is the enhanced equilibrium optimizer algorithm by SUN ET AL (2023), which improves the *Equilibrium Optimizer algorithm* (EO) of FARAMARZI ET AL (2020) with three additional communication strategies between particles. According to FARAMARZI ET AL (2020), EO is inspired by the principle of thermodynamics and pursues a state of equilibrium seen in physical systems. One solution algorithm for a multi-objective optimization problem is the hybrid adaptive differential evolution algorithm by WANG ET AL (2022), which randomly creates a population and improves it using a reverse learning strategy. Subsequently, mutations and crossovers are applied to create new individuals, followed by a selection process until the termination condition is satisfied.

The first investigations of assembly systems take place in the 1980s by GHOSH & GAGNON (1989). Generally, ALB und ALD primarily concentrate on the allocation of tasks to production stations in assembly lines (see MICHALOS ET AL 2015). As an extension of ALB, the ALD considers additional aspects of planning and layout from assembly lines. Typical assumptions for ALB are characterized by BOYSEN ET AL. (2007) and CHUTIMA (2022). For example, while the tasks in BOYSEN ET AL. (2007) only can be conducted in a specific way and the allocation of these tasks is only limited by the precedence matrix, every station has the same equipment and there are no parallel elements in assembly line. Regarding to Robotic Assembly Line Balancing, a specific form of ALB with focus of Robots in assembly line, CHUTIMA (2022) defines assumptions that each station has only one robot, unproductive times for e.g. positioning as well as time spent on loading, deloading and transporting parts and the costs of robots are negligible. This approach does not make all of these simplifying assumptions, but aims to model the real problem more realistically. Here, to date, no known approach deals with the non-discrete assignment of tasks to stations. With regards to the criteria of the objective function, most approaches optimize either the number of stations (e.g. DIDDEN ET AL (2023), LI ET AL (2023)) or the resulting production costs (e.g. FURUGI (2022), GUO ET AL (2022)). A dominat area of current research addresses the solving of multi-objective optimization problems (e.g. DIDDEN ET AL (2023), KANG & LEE (2023), CHEN ET AL (2023)). A frequently used solution approach for solving these optimization problems is the genetic algorithm. On the whole, the genetic algorithm comprises the following steps according to SRINIVAS & PATNAIK (1994): initialization and evaluation of an initial population, selection of a new generation, adding new solutions to the new generation by creation solutions through crossover and mutations, evaluation of the new generation, and repetition until the termination condition is met. Taking a closer look reveals individual differences. The algorithm of DIDDEN ET AL (2023) contains decoding at the end of the algorithm, which is responsible for the assignment of jobs and converts the created individual into an assembly line. KANG AND LEE (2023) applies the weighting method to transform multiple objective functions into one large objective function by generating weight with fuzzy analytic hierarchy process and extent analysis method, assigning a weight to each objective function and combining all objective functions into one objective function. Another adaptation of the genetic algorithm is the use of various initial methods to generate the initial population in LI ET AL (2023). CHEN AND JIA (2022) characterizes the genetic algorithm by randomly selecting change operator and an optional repair strategy. One algorithm for solving a combined problem of ALB and part feeding is the *Nested Bi-Level Multi-Objective Genetic Algorithm* (NSGA) in CHEN ET AL (2023), that integrates two genetic algorithms on different levels to optimize the ALB on the upper level and serves as the starting point for the part feeding on the lower level. Next, GUO ET AL (2022) integrates the variable neighborhood search method with NSGA-II to ensure the diversity of population. Moreover, a combination of NSGA-II with reinforcement learning is possible in ZHANG ET AL (2023), incorporation a Q-learning based strategy for selecting the best operator in each iteration. In addition of heuristic genetic algorithm, solving optimization problems with Bender´s decomposition method, proposed by BENDERS (1962), is an alternative that splits the original problem into a master problem and a slave problem, facilitating the iterative process of finding an optimal solution. Solving the master problem first to get a preliminary assignment of the processes serves to limit and further decompose the sequencing problems on the individual production cells of the slave problems in FURUGI (2022). Another approach of implementation according to HUANG ET AL (2022) is to modify Benders cuts by developing a sequence-based enumerative search method to compute effective combinatorial Bender cuts. Moreover, the branch and bound method according to NICKEL ET AL (2023) is a further optimal algorithm and branches the original problem into some simpler-to-calculate subproblems. HAGEMANN (2022) uses branch and bound method to solve ALD. Similarly, the ϵ-constraint-algorithm according to ABDOUS ET AL (2022) divides the original multi-objective problem into several subproblems with one objective function and treats the remaining goals as constraints. Defining different bounds for certain target functions, so called ϵ-values, for each subproblem causes different trade-offs between the objectives. Typical use-cases are found in the automotive industry, such as the final assembly of cars in DIDDEN ET AL (2023) and KANG & LEE (2023), as well as the assembly of car bodies in HAGEMANN (2022) and in the aerospace industry, examples include the assembly of aircraft in MAS ET AL (2016) and the assembly of aircraft wheel in MURA & DINI (2022).

To sum up, on the basis of a comprehensive literature analysis, the object of research and the current state of the art in ALD were identified. The resulting research gap raises the following three research questions, which will be answered in the subsequent phases of the DRM and are summarized below:

II. How can the problem of assigning tasks to stations be modeled realistically in the context of production system planning?
III. How can the mathematical model be set up and initialized and how can a solution algorithm be implemented?
IV. How can the approach be evaluated through an application with a real example from industry?

II. Modelling

In order to obtain a detailed understanding of the problem at hand, expert interviews were conducted and the current state of the art in terms of production system planning with Excel at the application company was analyzed. The general problem can be categorized according to BOYSEN ET AL (2007) as the assignment of tasks to stations for a given cycle time with the aim of minimizing the number of stations. However, in order to cover several optimization goals as well as all manufacturing and assembly processes and thus a divisibility of the tasks (as is usual for e.g. joining), the complex problem was presented in SCHAEFER ET AL (2023b). Generally, the common assumptions articulated by BOYSEN ET AL. (2007) and CHUTIMA (2022) are applicable in this context, albeit with some notable distinctions. Unlike traditional approaches, our model incorporates crucial factors such as loading and unloading durations for parts, alongside accounting for unproductive periods like positioning. Furthermore, we integrate the expenses linked to the utilization of robots into our modeling framework. Notably, our model offers a novel perspective by introducing the concept of task divisibility. To highlight only some aspekts of the model, the following provides a brief overview by first introducing relevant (decision) variables and parameters that will be utilized.

Table 1 – Model variables and parameters

(decision) variables:	Parameters
$anz_{R,j}$: number of handling robots on station j	A_{BR}: area of a processing robot
$c_{I,j}$: other variable costs	A_{ges}: available area
c_j: variable costs	A_M: area of a worker
$c_{M,j}$: labour costs	A_{OC}/A_{TC}: area of an orbit cell / a turntable cell
	C_{HR}/C_R: fixed costs of a handling/processing robots

$c_{S,j}$: electricity costs
d_j: allocate turntable to station j
f_j: fixed costs
$n_{R,j}$: number of processing robots on station j
$t_{b,j}$: processing time of station j
$t_{h,j}$: handling time of station j
$t_{z,j}$: cycle time of station j
x_{ij}: allocate parts of process i to station j
y_{jk}: allocate station type k to station j

C_{oc}/C_{TC}: fixed costs of an orbit cell / a turntable cell
F_{ik}: process task i with station type k
F_Q: factor
I: set of tasks
J: set of stations
$M(n,m)$: precedence graph
N: demand of product
O: OEE
P_{iu}: task i needs part u
Q_u: parts u with high dimensional accuracy
T_D: turning time of turntable
U: set of parts
$X_{min,k}$: minimum processing of tasks of type K

One of the two objective functions (1) minimizes the costs and comprises (2) fixed costs, (3) variable costs and a penalty term for quality. Each station item (robots, worktable) causes fixed costs, while variable costs reflect the labour pay, electricity costs and other variable costs, such as maintenance, influenced by (4) the cycle time of each station. The other objective function (5) focuses on flexibility by minimizing the utilized area of all items (worktable, robots, workers) from the stations.

$$min \sum_{j\in J} f_j + \sum_{j\in J} c_j + \sum_{i\in I}\sum_{j\in J}\sum_{u\in U} \left((|J| - j)^2 * \left(x_{ij} * P_{iu} * Q_u\right)\right) * F_Q \quad (1)$$

$$f_j = anz_{R,j} * C_{HR} + n_{R,j} * C_R + d_j * C_{TC} + (1 - d_j) * C_{oc} \quad (2)$$

$$c_j = (c_{M,j} + c_{S,j} + c_{I,j}) * N * \frac{1}{O} (in\text{ €}) \quad (3)$$

$$t_{z,j} = d_j * \left(max\left(t_{b,j}, t_{h,j}\right) + T_D\right) + \left(1 - d_j\right) * \left(t_{b,j} + t_{h,j}\right) \forall j \in J \quad (4)$$

$$min \sum_{j\in J} d_j * A_{TC} + \left(1 - d_j\right) * A_{OC} + (n_{R,j} - 1) * A_{BR} + anz_{HR,j} * A_{HR} + anz_{M,j} * A_M \quad (5)$$

In addition to the multi-criterial objective function, the constraints include. (6) allocation only one type for each station, (7) completing the processing of tasks over all stations, (8) considering the capabilities of stations for processing tasks, (9) adhering to a precedence graph, (10) complying with a minimum processing share for a process assignment depending on process type and (11) ensuring that the available space is not exceeded with allocated items over all stations.

$$\sum_{k\in K} y_{jk} \leq 1 \; \forall j \in J \quad (6)$$

$$\sum_{j\in J} x_{ij} = 1 \; \forall i \in I \quad (7)$$

$$\left(1 - y_{jk}\right) * x_{ij} + F_{ik} * (y_{jk})^T \geq x_{ij} \; \forall j \in J \; \forall i \in I \; \forall k \in K \quad (8)$$

$$0 = \sum_{j=0}^{r} x_{mj} * (1 - \sum_{j=0}^{r} x_{nj}) * M(n,m) \; \forall i = n,m \in I \; \forall r \in J \quad (9)$$

$$x_{ij} \geq y_{jk} * X_{min,k} * \lceil x_{ij} \rceil \; \forall i \in I, \forall j \in J, \forall k \in K \quad (10)$$

$$\sum_{j\in J} d_j * A_{TC} + (1 - d_j) * A_{OC} + n_{R,j} * A_{BR} + anz_{HR,j} * A_{HR} + anz_{M,j} * A_M \leq A_{ges} \quad (11)$$

III. Implementation

This paper considers the heuristic solution of a simplified problem as a preliminary study: For this, an example from industry was used to demonstrate the applicability of the approach.

The manufacturing company at hand is a Tier-1 automotive supplier that has to plan various customer variant-specific production lines in a short space of time under high cost pressure in order to manufacture and assemble components for many years at a high production volume. The product under consideration is a so-called rear twist beam, i.e. a rear axle that is welded together from several sheet metal parts. An initial cost estimate for the customer is currently being carried out manually in Excel. The planning process and the product are introduced in more detail by SCHAEFER ET AL (2022) respectively (2023a). The initialization of the OR-model and the solution space using the product and production system specifications is summarized in the following section.

The numerical initalization of a product containing 17 parts, 27 tasks, a given precedence graph, handling times, costs, weights etc. shows which information is coded and how. The time required to handle small and light parts is less for humans than for robots, while handling larger and heavier parts takes longer. In contrast, the situation is exactly the opposite with handling robots. Depending of the task type, a different level of minimum processing is set to avoid an unrealistic fragmentation of tasks. If a task is divided on two or more stations, then the processing time of the task increases by a value each time it is split up, as additional effort arises, for example due to an overlap of weld seams. Each station is a combination of handling robots, processing robots, workers, turntable and worktable according to a system construction kit. These items cause fix costs because of its purchase and installation and variable costs include electricity, worker´s pay, maintenance of robots and further factors. Therefore, they determine the first objective function, that minimizes the total costs of all stations. In context of the multi-objective optimization problem, they also affect the area and thus the second objective function, which minimizes the claimed area of all stations.

The NSGA-II algorithm is implemented using the Python library *Distributed Evolutionary Algorithms in Python* (DEAP) according to FORTIN ET AL (2012). Additional functions such as mutations (influencing decision variables such as number of stations, process allocation, handling options etc.) and repair mechanisms (ensuring constraints such as time, space, human ability etc.) have been added to the initial algorithm (grant permissible solutions, ensure constraints) according to Algorithm 1.

Algorithm 1: pseudo code of initial algorithm

```
Result: permissible individual
while a process has not yet been fully processed do
    determine the processes that have not yet been completed
    determine the processes that are not restricted by precedence graph
    determine the processes that are not already assigned to the current stations
    select randomly a share from a process that meets these requirements
    if the current station still has time capacity to process the selected share and has the right type
     then
        Assign the share to the station and update time capacity
    else
        try another share from a process according to the upper part
        if the current station still has not time capacity to process the selected share then
            Open a new station of a type to process unfinished processes
```

Algorithm 1 – Initial algorithm

The program flow is as follows: It starts by randomly generating an initial population using algorithm I and then evaluating the generated individuals using the objective functions. In each subsequent generation, the population is first modified using ten different mutations. When a modified individual violates a constraint, a repair mechnism corrects it. This is followed by an assessment of the individuals in the modified population. The original population and the modified population are combined to form a new population. Selection operators of NSGA-II, which implements DEAP, select the individuals of the new generation, which forms the basis for the next generation.

IV. Findings

After generating different solutions with the initial algorithm for the first generation, the best solution of each generation is improved (by mutation/repair) until convergence, as seen in Fig. 2-l. Refer to Fig. 2-r for the Pareto front, the solutions of which are described below.

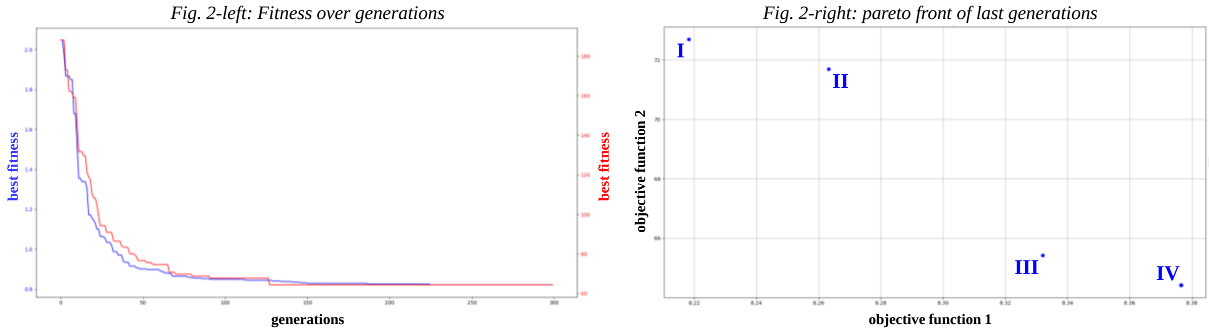

Figure 2 – Results of genetic algorithm

Fig. 3-a shows the characteristics of the best permissible solution according to the first objective function (minimizing cost) in the first generation, that is generated by the initial algorithm. It illustrates task allocation to the stations and their resulting handling times. However, there are options to minimize costs and achieve a better result, for example, by reducing the number of stations and using stations more efficiently. As a result of the algorithm, Fig. 3-b depicts the best solution according to the first objective function in the last generation (solution I in Fig. 2-r), where the costs of stations have been reduced.

Of the 8.22 million euros of the first objective function value in solution I, 1.64 million euros is attributable to fixed costs and 6.58 million euros to variable costs. Stations 1, 2, 4, 5 and 6 are responsible for welding processes. The third station carries out laser cutting processes, while the last station is responsible for quality testing. One station uses two processing robots to halve the processing time because the variable costs decrease more than the fixed costs of an additional processing robot. As the number of processes and the processing time of a station increase, the likelihood of the algorithm assigning two processing robots to that station also increases. It should be noted that stations 1 and 3 each have only one processing robot. Even though turntables parallelize processing and handling actions to reduce the cycle time of a station and therefore the variable costs, no stations have a turntable because of the huge fixed costs of a turntable and the additional costs for use and maintenance are higher than the reduction of variable costs. In general, splitting a task leads to a longer processing time due to the additional effort and thus to higher variable costs, which is why this solution does not include split processes. Due to the weight of the individual parts, which already exceeds the maximum carrying weight of a person at the first station, it is not possible in the production system for a person to unload and transfer them to the next station. Loading the stations with individual parts by humans is theoretically possible, but due to the high variable labour costs, the algorithm decides that robots should load all stations.

The same observation can be made for the best solution according to the second objective function i.e. minimizing area (solution IV in Fig. 2-r). Of the 64 square metres of the second objective function value in solution IV, 49 square metres is attributable to stations, three square metres to three additional processing robots and 12 square metres to six handling robots. Welding processes takes place in stations 1, 2, 4 and 5, while the third station is dedicated to laser cutting processes. The final station is responsible for conducting quality testing. As seen in Table 1, the number of stations is even fewer than with the solution of Fig. 3-b because one task is divided and allocated to two stations, and the utilization of the other stations is higher because of an efficient allocation of the tasks. The turntable at the fourth station enhances processing efficiency within a customer cycle by parallelizing processing and handling and helps to reduce the number of stations with rising costs. Although the algorithm can assign additional handling robots to a station in order to reduce the handling time and create capacities for further processes, the algorithm, similar to solution I, does not avoid the assignment to further handling robots at any station due to increasing costs and space requirements.

Fig. 3-a: Assignment of tasks to stations of first generation

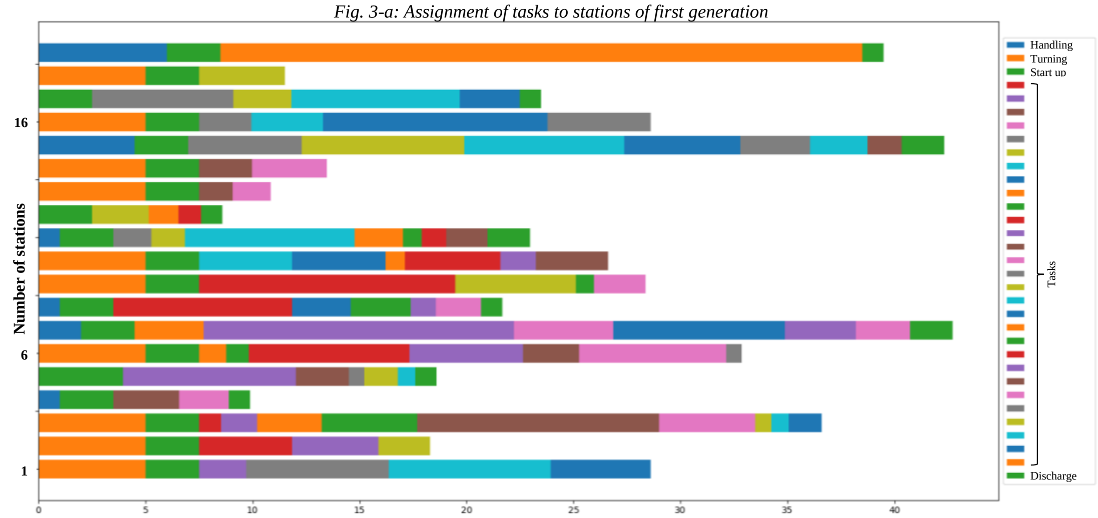

Fig. 3-b: Assignment of tasks to stations of last generation

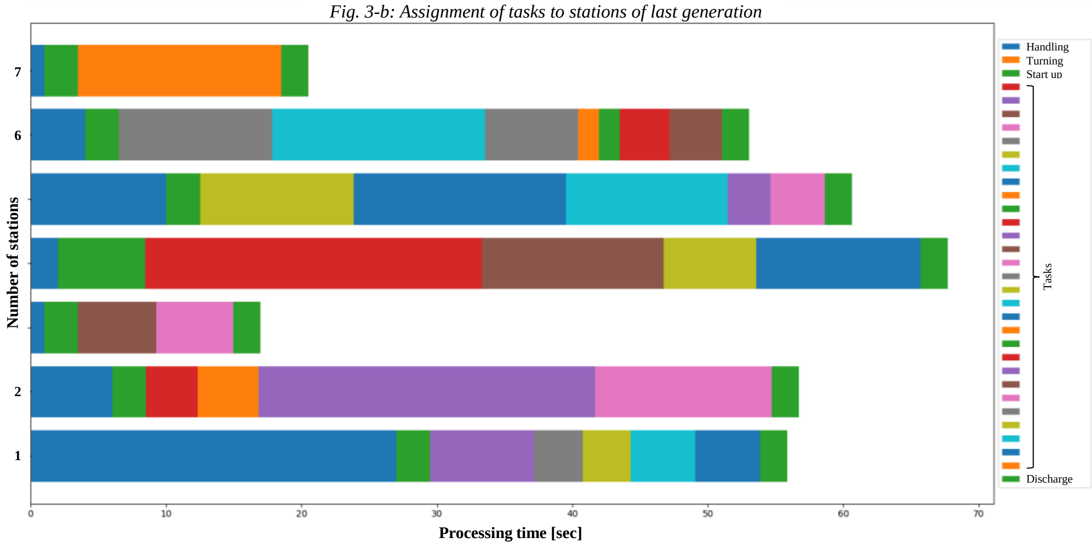

Figure 3 – Assignment of tasks to stations

All findings of each solution from Fig. 2-r are summarized in Table 2. Additionally, it includes the degree of objective achievement of the two other solutions (II & III) from Fig. 2-r, which represent trade-offs between the objective functions.

Table 2 – Summary of solutions

		amount of production cells	costs [Mill. €]	area [m^2]
single-objective	solution I	7	8,2182	72,70
	solution IV	6	8,3765	64,42
	optimum	6	8,2182	64,42
multi-objective	solution II	7	8,2632	71,70
	degree of objective achievement	85,71%	99,46%	89,85%
	solution III	6	8,3320	65,42
	degree of objective achievement	100%	98,63%	98,47%

Even though the NSGA-II algorithm addresses the issue of the local optima with crowding distance, the repeatability of the solution is limited. A sensitivity analysis with the number of population and the number of generations shows that the algorithm achieves better solutions with an increasing number for one of these parameters, up to a certain point, as illustrated in figure 4. Even at the beginning, with low values for both hyperparameters, the algorithm achieves significant improvements with small increases. If the values for both hyperparameters are high, further improvement through increasing values cannot be achieved. A direct comparison of both hyperparameters shows that the number of generations improves the results of the algorithm more than the population size. This can be seen in figure 4, where the right side has a rapidly falling curve when compared to the higher values of the left side.

Fig. 4-left: Sensitivity analysis (obj. func. 1) *Fig. 4-right: Sensitivity analysis (obj. func. 2)*

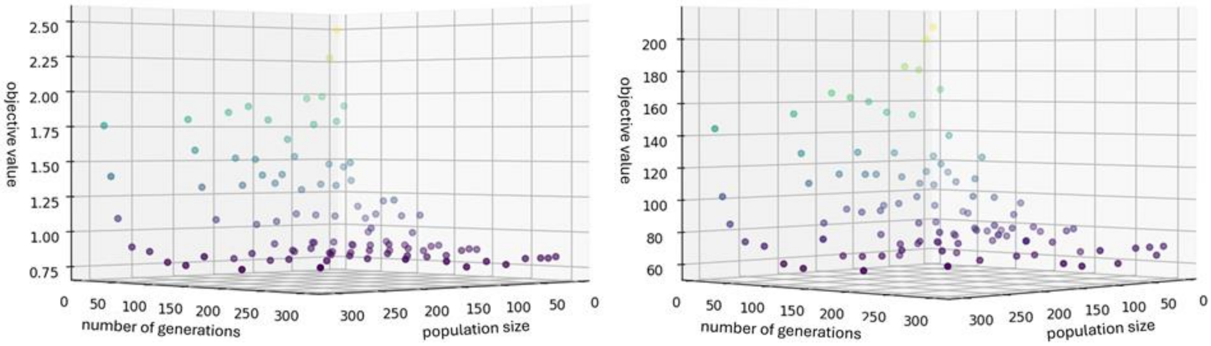

Figure 4 – Sensitivity analysis of population size and number of generations

Relevance & Contribution to Research and Practice

The findings show the application of an OR model to a real-world example and indicate the benefits of assisted planning and comparability between different scenarios compared to manual planning with Excel. In age of Industry 4.0, the giving method can contribute to digitalize the planning process and accelerate the efficient allocation of ressources in assembly line planning. Explicitly considering various objectives in a more complex world offers additional benefits. The purpose of our approach is to support the production system planning by producing rough structures of the final design. Although the mathematic model underlies some assumptions and the algorithmus has some limitations, the model presented offers a more realistic representation of the problem by incorporating widely accepted assumptions. Looking at current research, modeling divisible tasks represents a novel aspect of the assembly line design problem. As seen in the findings, the divisibility of tasks is a useful extension of modeling. Finally, the work is a central basis for planning a circular factory. The assistance system and the use of genetic algorithms are advantageous here, as the optimization of the system is based on domain knowledge and constant reconfiguration.

Relevance & Contribution to Research and Practice

Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – SFB-1574 – 471687386.

References

Abdous, M.-A., Delorme, X., Battini, D. & Berger-Douce, S. (2022), Multi-objective collaborative assembly line design problem with the optimisation of ergonomics and economics, International Journal of Production Research.

Álvarez-Miranda, E. & Pereira, J. (2019), On the complexity of assembly line balancing problems, Computers & Operations Research, Vol. 108.

Benders, J. F. (1962), Partitioning procedures for solving mixed-variables programming problems, Numerische Mathematik, Vol. 4, Nr. 1.

Blessing, L. T. M. and Chakrabarti, A. (2009), DRM, a Design Research Methodology, Springer, London.

Boysen, N., Fliedner, M., Scholl, A. (2007), A classification of assembly line balancing problems, European Journal of Operational Research 183 (2), pp. 674–693.

Chen, J. & Jia, X. (2022), Energy-efficient integration of assembly line balancing and part feeding with a modified genetic algorithm, The International Journal of Advanced Manufacturing Technology, Vol. 121, Nr. 3-4.
Chen, J., Jia, X. & He, Q. (2023), A novel bi-level multi-objective genetic algorithm for integrated assembly line balancing and part feeding problem, International Journal of Production Research, Vol. 61, Nr. 2.
Chutima, P. (2022), „A comprehensive review of robotic assembly line balancing problem“, in: Journal of Intelligent Manufacturing, Bd. 33, Nr. 1, S. 1–34.
Dalle Mura, M. & Dini, G. (2022), Job rotation and human–robot collaboration for enhancing ergonomics in assembly lines by a genetic algorithm, The International Journal of Advanced Manufacturing Technology, Vol. 118, Nr. 9-10.
Didden, J. B. H. C., Lefeber, E., Adan, I. J. B. F. & Panhuijzen, I. W. F. (2023), Genetic algorithm and decision support for assembly line balancing in the automotive industry, International Journal of Production Research, Vol. 61, Nr. 10.
Faramarzi, A., Heidarinejad, M., Stephens, B. & Mirjalili, S. (2020), Equilibrium optimizer: A novel optimization algorithm, Knowledge-Based Systems, Vol. 191.
Fortin, F.-A., de Rainville, F.-M., Gardner, M.-A., Parizeau, M. and Gagne, C. (2012), DEAP: evolutionary algorithms made easy, The Journal of Machine Learning Research, Nr. 13.
Furugi, A. (2022), Sequence-dependent time- and cost-oriented assembly line balancing problems: a combinatorial Benders' decomposition approach, Engineering Optimization, Vol. 54, Nr. 1.
Ghosh, S. and Gagnon, R. J. (1989), A comprehensive literature review and analysis of the design, balancing and scheduling of assembly systems, International Journal of Production Research 27 (4) pp. 637–670.
Guo, J., Pu, Z., Du, B. & Li, Y. (2022), Multi-objective optimisation of stochastic hybrid production line balancing including assembly and disassembly tasks, International Journal of Production Research, Vol. 60, Nr. 9.
Hagemann, S. (2022), Algorithmische Konzeption von hochautomatisierten Fließmontagesystemen am Beispiel des automobilen Karosseriebaus.
Huang, D., Mao, Z., Fang, K. & Yuan, B. (2022), Combinatorial Benders decomposition for mixed-model two-sided assembly line balancing problem, International Journal of Production Research, Vol. 60, Nr. 8.
Kang, H.-Y. & Lee, A. H. I. (2023), An evolutionary genetic algorithm for a multi-objective two-sided assembly line balancing problem: a case study of automotive manufacturing operations, Quality Technology & Quantitative Management, Vol. 20, Nr. 1.
Kirkpatrick, S., Gelatt, C. D. & Vecchi, M. P. (1983), Optimization by simulated annealing, Science (New York, N.Y.), Vol. 220, Nr. 4598.
Li, S., Butterfield, J. & Murphy, A. (2023), A New Multi-Objective Genetic Algorithm for Assembly Line Balancing, Journal of Computing and Information Science in Engineering, Vol. 23, Nr. 3.
Mas, F., Ríos, J., Gómez, A. & Hernández, J. C. (2016), Knowledge-based application to define aircraft final assembly lines at the industrialisation conceptual design phase, International Journal of Computer Integrated Manufacturing.
Michalos, G., Fysikopoulos, A., Makris, S., Mourtzis, D. & Chryssolouris, G. (2015), Multi criteria assembly line design and configuration – An automotive case study, CIRP Journal of Manufacturing Science and Technology, Vol. 9.
Nickel, S., Rebennack, S., Stein, O. & Waldmann, K.-H. (2022), Operations Research. Springer Berlin Heidelberg and Springer Gabler, Berlin, Heidelberg, Vol. 3.
Schäfer, L., Frank, A., May, M. C. and Lanza, G. (2022), Automated derivation of optimal production sequences from product data, Procedia CIRP 107 (04) pp. 469–474.
Schäfer, L., Treml, N., May, M. C. and Lanza, G. (2023a), Classifying Parts using Feature Extraction and Similarity Assessment, Procedia CIRP, 119, pp. 944–949.
Schäfer, L., Kochendörfer, P., May, M.C. and Lanza, G. (2023b), Planning and Multi-Objective Optimization of Production Systems by means of Assembly Line Balancing, Procedia CIRP CMS 2023.
Srinivas, M. & Patnaik, L. M. (1994), Genetic algorithms: a survey, Computer, Vol. 27, Nr. 6.
Sun, Y., Pan, J.-S., Hu, P. & Chu, S.-C. (2023), Enhanced Equilibrium Optimizer algorithm applied in job shop scheduling problem, Journal of Intelligent Manufacturing, Vol. 34, Nr. 4.
van Laarhoven, P. J. M., Aarts, E. H. L. & Lenstra, J. K. (1992), Job Shop Scheduling by Simulated Annealing, Operations Research, Vol. 40, Nr. 1.
Wang, G.-G., Da Gao & Pedrycz, W. (2022), Solving Multiobjective Fuzzy Job-Shop Scheduling Problem by a Hybrid Adaptive Differential Evolution Algorithm, IEEE Transactions on Industrial Informatics, Vol. 18, Nr. 12.
Yang, X.-S. & Deb, S. (2009), Cuckoo Search via Lévy flights, in: 2009 World Congress on Nature & Biologically Inspired Computing, pp. 210–214. IEEE, [Piscataway, N.J.].
Zhang, Z., Tang, Q., Chica, M. & Li, Z. (2023), Reinforcement Learning-Based Multiobjective Evolutionary Algorithm for Mixed-Model Multimanned Assembly Line Balancing Under Uncertain Demand, IEEE transactions on cybernetics, Vol. PP.