

INTERNETBASIERTE DESIGNOPTIMIERUNG MIT HyGLEAM

Wilfried Jakob

Institut für Angewandte Informatik, Forschungszentrum Karlsruhe (FZK)
Email: jakob@iai.fzk.de

Dagmar Peters

Institut für Theoretische Elektrotechnik und Mikroelektronik, Uni Bremen (ITEM)
Email: peters@item.uni-bremen.de

Andreas Reiffer

InterConnect Software GmbH, Karlsruhe
Email: reiffer@interconnect.de

Kurzfassung

Mit der internetbasierten Variante von HyGLEAM können Optimierungen über einen Remoteaufruf auf einem entfernten Rechner eines Application Service Providers ausgeführt werden. HyGLEAM integriert zwei allgemein anwendbare lokale Suchverfahren in den global arbeitenden Evolutionären Algorithmus GLEAM und kombiniert die Vorteile der beteiligten Verfahren unter Vermeidung ihrer jeweiligen Nachteile. Dazu wird eine neue Methode zur Steuerung des Verfahrenseinsatzes benutzt. Die Robustheit und die gesteigerte Performance des resultierenden hybriden Verfahrens werden an Hand mathematischer Benchmarkfunktionen und dreier Designanwendungen nachgewiesen.

1 Einleitung

Das Verbundprojekt OMID (Optimierung von Mikrosystemen für Diagnose- und Überwachungsanwendungen) hat unter anderem zum Ziel, die im Vorgängerprojekt DEMIS entwickelten und getesteten Optimierungsverfahren zu integrieren, die dabei entstehenden Hybride zu bewerten und das resultierende Optimierungswerkzeug in einer webbasierten Remote-Version zur Verfügung zu stellen. In DEMIS wurde deutlich, dass alle untersuchten Verfahren ihre Stärken und Schwächen haben [1, 2]. So liefern lokale Suchverfahren wie der Rosenbrock-Algorithmus oder das Complex-Verfahren vergleichsweise schnell ein Ergebnis, das aber bei multimodalen Aufgabenstellungen meist nur ein lokales Optimum dar-

stellt. Globale Suchverfahren wie die Evolutionären Algorithmen (EA) bieten hingegen eine wesentlich größere Chance, das globale Optimum zu finden, benötigen dafür aber zum Teil wesentlich mehr Berechnungen der Qualitätsfunktion. Um die meist aufwendigen Simulationen zur Bestimmung der Qualität einer Designlösung zu verringern ohne den globalen Charakter der Suche aufzugeben, wurde versucht, die beiden Algorithmenklassen so geschickt miteinander zu verbinden, dass der resultierende Hybrid die Vorteile der beteiligten Verfahren ohne deren Nachteile in sich vereint. Erste Tests an Designoptimierungsaufgaben aus dem laufenden Projekt zeigten vielversprechende Ergebnisse [3]. In Kap. 2 werden die am FZK entwickelten Hybridisierungsmethoden [4, 5] vorgestellt. Integriert wurden zwei am ITEM implementierte lokale Suchverfahren [1] und das GLEAM-Verfahren (General Learning Evolutionary Algorithm and Method) [6, 7], das am FZK zu HyGLEAM (Hybrid General Purpose Evolutionary Algorithm and Method) weiterentwickelt und als GADO (Genetischer Algorithmus zur Designoptimierung) und HyGADO zur Designoptimierung eingesetzt wurde. Kap. 3 beschreibt die untersuchten Testfälle bestehend aus mathematischen Benchmarkfunktionen und OMID-Anwendungen. In Kap. 4 sind die Testergebnisse und die sich daraus ergebenden Schlussfolgerungen enthalten. Kap. 5 stellt die auf dem Orbit-System der Firma InterConnect [8] basierende Remote-Lösung von HyGLEAM vor, während Kap. 6 mit einer Zusammenfassung des Beitrags schließt.

2 Das HyGLEAM-Konzept

Das in [4, 5] ausführlicher dargestellte Konzept von HyGLEAM kann hier nur in der gebotenen Kürze erläutert werden. Zu den beiden lokalen Verfahren gibt Schwefel [9] eine ausführlichere Beschreibung mit weiteren Anwendungsfällen.

2.1 Benutzte Optimierungsverfahren

2.1.1 GLEAM

GLEAM ist ein eigenständiger evolutionärer Algorithmus, der Elemente der Evolutionsstrategie und der (reellcodierten) Genetischen Algorithmen mit Konzepten der Informatik (abstrakte Datentypen) verbindet. Die Codierung der zu optimierenden Parameter erfolgt in den sogenannten Aktionen. Eine Aktion besteht aus ein oder mehreren Parametern vom Typ Bool, Integer oder Real entsprechend ihrem anwendungsspezifisch definierten Aktionstyp. Eine Aktion drückt damit eine bestimmte Eigenschaft eines Designs aus, z.B. die Anwesenheit einer bestimmten Komponente, die durch die Aktionsparameter näher bestimmt wird oder die Relationen zu anderen Komponenten entsprechend den Aktionsparametern eingeht. Im einfachsten Fall codieren die Aktionen nur gewisse einzelne Parameter eines Designs. Die Aktionen bilden eine Aktionenkette (AK), die den Chromosomen des biologischen Vorbilds entspricht. Es gibt derzeit in GLEAM drei AK-Typen: Bei den ersten beiden ist jeder Aktionstyp mit genau einer Aktion vertreten und sie unterscheiden sich darin, ob die Reihenfolge der Aktionen relevant ist. Die Variante ohne bedeutungstragende Reihenfolge wird bei Designoptimierungsaufgaben, bei denen es um reine Parameteroptimierung geht, verwendet. Die andere Variante spielt bei OMID keine Rolle, sie kommt aber bei anderen Anwendungen wie z.B. Schedulingproblemen zum Einsatz [10]. Schließlich gibt es noch den AK-Typ mit variabler Länge, bei dem jeder Aktionstyp mit keiner, einer oder mehreren Aktionen in beliebiger Reihenfolge vertreten sein kann. Dieser Typ findet bei anspruchsvolleren Designoptimierungsaufgaben wie z.B. der in Kap. 3 beschriebenen Aktorplatte Verwendung.

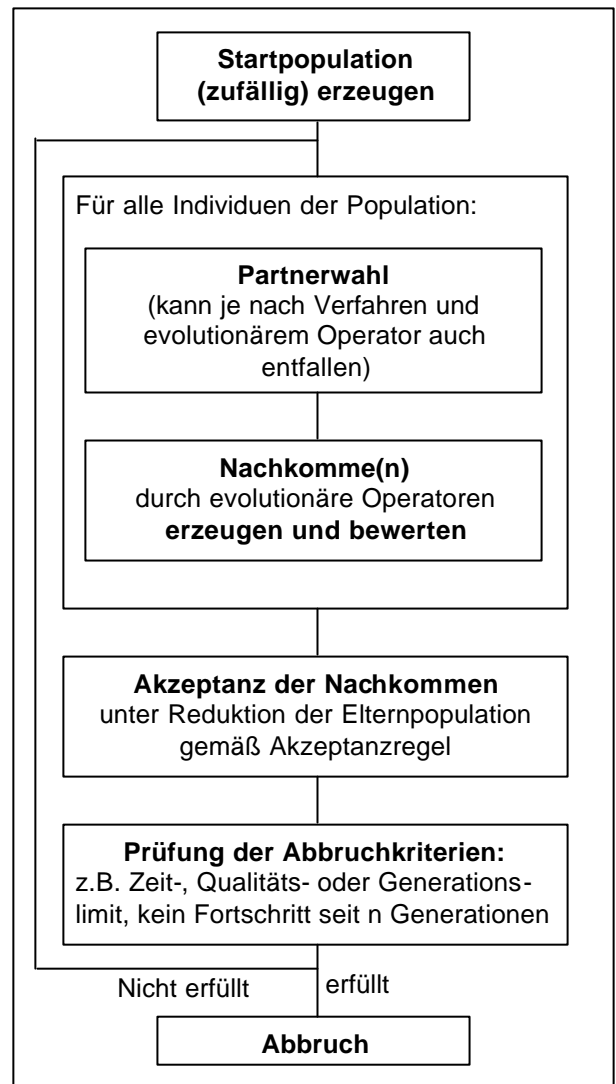


Abb. 1: Prinzipieller Ablauf Evolutionärer Algorithmen

Abb. 1 zeigt den prinzipiellen Ablauf Evolutionärer Algorithmen. Ausgehend von einer meist zufällig erzeugten Startpopulation, die durch manuell oder auf andere Weise erzeugte Lösungen angereichert werden kann, werden alle Individuen generationsweise der nachgebildeten Evolution unterworfen. Dazu werden Nachkommen durch Mutation und/oder durch Rekombination des Genmaterials zweier Eltern erzeugt. Im letzten Fall findet zuvor eine Partnerwahl statt, die meist auf der Fitness der Individuen beruht. In GLEAM wird dafür eine ranking-basierte Selektion benutzt, die die Chancen der besten Individuen, selektiert zu werden, etwas zu Gunsten der schlechteren dämpft. Dadurch soll unter anderem vorzeitiger Konvergenz auf ein Suboptimum vorgebeugt werden. GLEAM enthält einen Satz vordefinierter genetischer Operatoren (Muta-

tionen der Werte oder der Aktionsposition, verschiedene Crossover-Operatoren), die aufgabenspezifisch konfiguriert zur Anwendung gebracht werden können. Die Wertemutationen berücksichtigen die in den Aktionstypdefinitionen mithinterlegten Wertebereichsgrenzen, sodass explizite Restriktionen immer eingehalten werden. Kleine Änderungen sind dabei entsprechend der Evolutionsstrategie wahrscheinlicher als große. Pro Paarung werden in GLEAM mehrere Nachkommen erzeugt, wobei die dabei beteiligten genetischen Operatoren sowie ihre Anwendungswahrscheinlichkeiten problemspezifisch ausgewählt und konfiguriert werden können. Schließlich müssen die Nachkommen der Elterngeneration akzeptiert oder verworfen werden. Die akzeptierten Kinder ersetzen ebensoviel Mitglieder der Elterngeneration, sodass die Populationsgröße konstant bleibt. In GLEAM wird dazu eine elitäre Akzeptanzregel benutzt, bei der das beste Individuum unabhängig von seinem Alter immer überlebt, sodass eine Verschlechterung der pro Generation erreichten Lösungsqualität nicht eintreten kann.

2.1.2 Verfahren nach Rosenbrock

Kernprinzip des Verfahrens nach Rosenbrock [11] ist die Suche des Optimums entlang der Richtungen eines im n -dimensionalen Raum rotierenden Koordinatensystems. Beschränkungen werden mit einer partiellen, internen Straffunktion berücksichtigt.

Das in Abb. 2 dargestellte Verfahren setzt einen zulässigen Startpunkt voraus. Im Anschluss an die Initialisierung wird ein Testschritt in die aktuelle Richtung i gemacht. Sind alle Bedingungen erfüllt und ist der Wert der partiellen Barrierefunktion ermittelt worden, wird untersucht, ob es sich bei dem aktuellen Punkt um einen besseren handelt als bei dem Vorgänger. Ist dies der Fall, vergrößert sich die Schrittweite um den Faktor drei. Wurde mit dem aktuellen Punkt kein Erfolg erzielt, wird die Suche in die entgegengesetzte Richtung mit halbiertem Schrittweite fortgesetzt. Innerhalb einer Iteration muss in jede Richtung des aktuellen Koordinatensystems mindestens ein Erfolg gefolgt von einem Misserfolg erzielt werden, bevor das Abbruchkriterium überprüft wird.

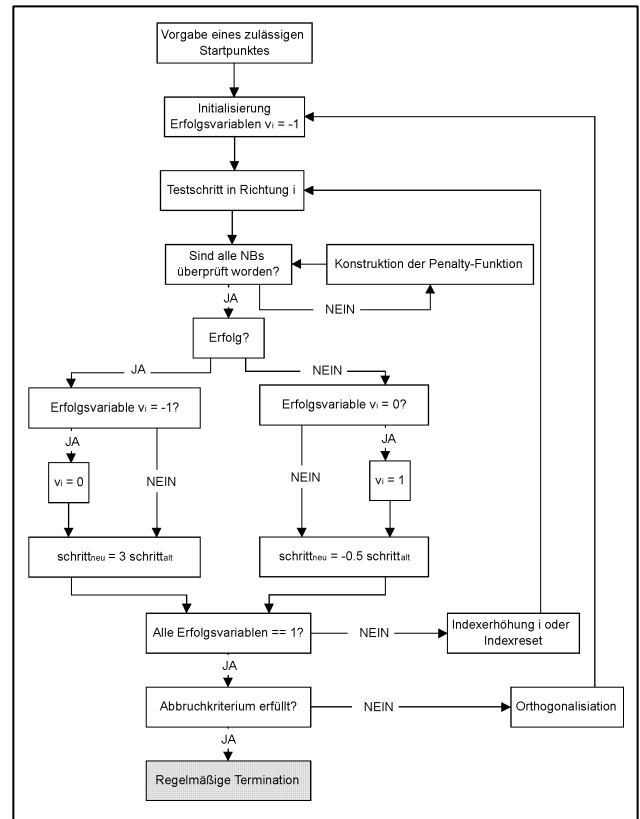


Abb. 2: Ablaufdiagramm des Rosenbrock-Verfahrens

Wird das Abbruchkriterium erfüllt, terminiert das Verfahren, ansonsten wird eine Rotation des Koordinatensystems durchgeführt. Dazu wird die vielversprechendste Richtung der letzten Iteration als Hauptachse des neuen Koordinatensystems festgelegt und die restlichen $n-1$ Achsen orthogonal zu dieser Achse aufgebaut. Mit diesem Koordinatensystem setzt sich die Suche des Optimums so lange fort, bis das Abbruchkriterium erfüllt wird.

2.1.3 Complex-Algorithmus

Das Complex-Verfahren wurde 1965 von Box [12] in Anlehnung an den Simplexalgorithmus für nichtlineare Funktionen von Nelder und Mead entwickelt. Der prinzipielle Ablauf ist in Abb. 3 dargestellt. Der Complex-Algorithmus wurde speziell für beschränkte Problemstellungen entwickelt und verwendet keinen der klassischen Ansätze: externe Straffunktion, interne Straf- bzw. Barrierefunktionen, Lagrange-Multiplier-Verfahren oder partielle Straffunktion.

Ausgehend von einem zulässigen Startpunkt, wird der Startcomplex gebildet. Es folgt die Ermittlung des schlechtesten Punktes sowie

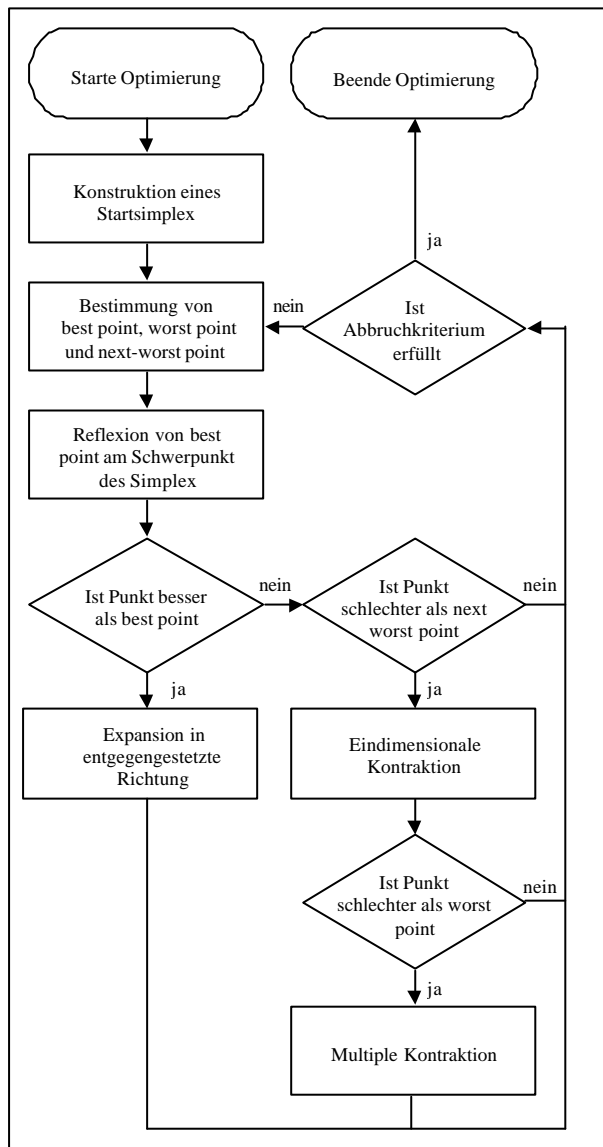


Abb.3: Prinzipieller Ablauf von Polyederverfahren

des Scheitelpunktes der verbleibenden Punkte. Ist der Scheitelpunkt zulässig, wird der schlechteste Punkt an ihm reflektiert. Dabei handelt es sich um eine Normalreflexion, bei der gleichzeitig eine Expansion des Complex durchgeführt wird. Bei Nichtzulässigkeit des Scheitelpunktes wird ein neuer Startcomplex gebildet und das Verfahren neu gestartet. Auch der reflektierte Punkt muss daraufhin überprüft werden, ob er alle formulierten Bedingungen erfüllt. Ist dies nicht der Fall, wird der Reflexionsfaktor halbiert und eine erneute Reflexion durchgeführt, bis ein zulässiger Punkt gefunden wird oder der Reflexionsfaktor kleiner als eine bestimmte untere Grenze geworden ist. In diesem Fall wird der Reflexionsvorgang mit dem nächst schlechtesten

Punkt wiederholt. Kann für keinen Punkt des Complex ein zulässiger Reflexionspunkt gefunden werden, bricht das Verfahren vorzeitig ab. Ein zulässiger Reflexionspunkt wird daraufhin untersucht, ob mit ihm ein verbesserter Wert der Zielfunktion erzielt wird. Ist dies der Fall, wird das Abbruchkriterium überprüft; ist es erfüllt, terminiert das Verfahren regelmäßig, ansonsten wird der schlechteste Punkt ermittelt und die nächste Iteration des Optimierungsprozesses gestartet.

2.2 Hybridisierungsarten

Die Hybridisierung beruht auf folgenden vier Ansätzen:

- **Initialisierung der Startpopulation**
Die Startpopulation wird ganz oder teilweise mit den Ergebnissen von Läufen des lokalen Verfahrens initialisiert. Ein eventueller Rest wird zufällig bestimmt. Der Grundgedanke besteht darin, die evolutionäre Suche bereits mit Lösungen einer gewissen Qualität beginnen zu lassen und so die Zeit zum Auffinden befriedigender Bereiche des Suchraums zu sparen.
- **Nachoptimierung der GLEAM-Ergebnisse**
Die Nachoptimierung der GLEAM-Ergebnisse mit konventionellen Verfahren wird häufig auch als „*Local Hill Finding + Local Hill Climbing*“ bezeichnet. Man benutzt den EA zum Auffinden einiger vielversprechender Regionen des Suchraums und verwendet anschließend ein lokales Verfahren zur Bestimmung des exakten (lokalen) Optimums. Das beste dieser Optima ist dann die Lösung. Auf die Frage, wann der EA sinnvoll beendet und die lokale Suche begonnen werden kann, wird im nächsten Abschnitt eingegangen.
- **Direkte Integration**
Hierbei werden alle oder nur der beste Nachkomme einer Paarungsoperation von GLEAM mit dem lokalen Verfahren optimiert. Damit operiert GLEAM quasi nur noch über den Bergspitzen der Fitnesslandschaft.
- **Verzögerte Direkte Integration**
Bei vielen Aufgabenstellungen bringt eine lokale Optimierung ausgehend von

schlechten Startwerten meist nicht viel. Daher erscheint es sinnvoll, die direkte Integration erst nach einer anfänglichen Phase evolutionärer Suche dazuzuschalten.

Zu diesen vier Hybridisierungsarten kommen noch die Kombinationen von Voroptimierung und (verzögerter) direkter Integration. Bei der Nachoptimierung mit dem Complex wurden zwei Arten untersucht: Entweder wurden alle von GLEAM gelieferten Lösungen zur Bildung eines einzigen Start-Complexes genutzt oder sie dienten einzeln als Startpunkte für entsprechend viele Complex-Läufe. Daraus ergaben sich mit den beiden lokalen Verfahren 13 verschiedene Hybridisierungsarten.

2.3 Steuerung der Hybridisierung

Die auf dem OMID-Statusseminar [3] bereits ausführlich dargestellten Verfahren zur Steuerung der Hybridisierung beruhen auf der genotypischen Varianz in der Population. Diese wird nur bei auftretender Fitness-Stagnation bestimmt, um Rechenzeit zu sparen. Wenn es in der Population weniger als N_{max} Nischen von Individuen gibt, deren genotypische Differenz geringer als ein vorgegebenes ϵ ist und alle Nischen eine geringere Differenz als ϵ_{Pop} aufweisen, gilt die Population als „konvergiert“. Dann wird die Evolution bei der Nachoptimierung abgebrochen und alle Nischen

steuern einen Vertreter zum Gesamtergebnis bei, das noch um den Populationsbesten ergänzt wird, sofern er besser als alle Nischenvertreter ist. Diese Individuen werden mit dem jeweiligen lokalen Verfahren nachoptimiert und das beste ist schließlich das Endergebnis.

Im Falle der verzögerten direkten Integration wird das Konvergenzkriterium weniger streng als bei der Nachoptimierung parametrisiert. Es bestimmt den Zeitpunkt der Zuschaltung der lokalen Suche.

3 Testfälle

Untersucht wurden zusätzlich zu den in DEMIS [1] bereits benutzten Testfällen noch die Testfunktion nach Fletcher und Powel [9, Probl. 2.13] und die im Statusseminar [3] bereits behandelten Anwendungen aus dem laufenden Projekt. Tab. 1 zeigt wichtige Eigenschaften der Testfälle und gibt die Ergebnisse für die einzelnen Verfahren an. Alle Experimente wurden mit vergleichbaren Einstellungen der beteiligten Verfahren durchgeführt. Variiert wurden wie in [3] dargestellt lediglich beim Rosenbrock die Abbruchschranke und bei GLEAM die Populationsgröße. Abweichungen von den in [1] publizierten Ergebnissen erklären sich bei den beiden lokalen

Experiment	Parame-teranz.	Modalität	Implizite Restrikt.	Genau-igkeit	Erfolgsrate [%] und Aufwand [Evaluationen]					
					GLEAM		Rosenbrock		Complex	
Bracken	2	multimod.	ja	1,0E-03	100	2509	4	1074	70	61
				1,0E-06	100	5000	-	-	98	196
Beale	3	unimod.	ja	1,0E-03	100	28936	37	4854	100	164
				1,0E-06	100	76229	2	1538	98	269
Bigg	5	multimod.	nein	1,0E-03	100	3370501	17	4001	20	164
				1,0E-06	-	-	18	18544	10	799
Helical Valley	3	unimod.	nein	1,0E-06	100	47288	71	15062	-	-
Rastrigin	20	multimod.	nein	1,0E-04	100	51721	-	-	-	-
Fletcher & Powel	5	multimod.	nein	1,0E-05	100	483566	10	1090	10	243
Fraktale Fkt.	20	multimod.	nein	Ziel: -0.05	100	195100	-	-	-	-
Heterodyn-Empf.	3	multimod.	nein		100	5773	8	94	12	102
Drucksensor	3	multimod.	nein		100	29791	-	-	-	-
Aktorplatte	dynam. n x 4	multimod.	ja							

Tabelle 1: Wichtige Eigenschaften der Testfälle und Ergebnisse der drei Verfahren

Verfahren durch den Verzicht, weitere Parameter wie z.B. die initiale Schrittweite anzupassen. Bei der Testfunktion nach Bracken wurde bei GLEAM (GADO) seinerzeit versehentlich der Aufwand zum Auffinden des exakten Optimums anstelle einer 10^{-3} -Genauigkeit angegeben.

In den drei Ergebnisspalten von Tab. 1 wird die Erfolgsrate in Prozent und der durchschnittliche Aufwand in Evaluationen basierend auf jeweils 100 Läufen für die jeweils beste Parametrierung angegeben. Ein – bedeutet, dass das betreffende Verfahren nicht in der Lage war, das Problem unter den gegebenen Einstellungen zu lösen. Beim Rosenbrock wurden folgende normierte Abbruchschranken verwendet: 0.1 (s), 0.01 (n), 10^{-4} (m), 10^{-6} (h) und 10^{-8} (x) und bei GLEAM wurde die Populationsgröße variiert. Ab der fraktalen Funktionsind in der Tabelle keine Genauigkeitsangaben vorhanden, da die exakte Lösung nicht bekannt ist. Bei der Aktorplatte handelt es sich auf Grund Ihrer aufwendigen Simulation um eine „echte Anwendung“ in dem Sinne, dass keine Läufe zum Verfahrensvergleich möglich sind, sodass hier nur die Eigenschaften gelistet sind.

Der Heterodynempfänger [2,13] ist eine Komponente eines mikrooptischen Kommunikationssystems bestehend aus zwei Kugellinsen und einer Photodiode. Der Abstand zwischen dem Lichtleiter und erster Kugellinse sowie die beiden Brechungsindizes sollen so bestimmt werden, dass nicht nur eine optimale Ausleuchtung der Photodiode im Kollimationspunkt erreicht wird, sondern die ganze Anordnung auch möglichst unempfindlich gegenüber Fertigungstoleranzen innerhalb vorgegebener Grenzen ist. Letzteres macht die ansonsten einfache Aufgabe zu einem trotz der nur drei Parameter schwierigen, stark multimodalen Optimierungsproblem.

Die Wandlung der mechanischen Größe Druck in ein elektrisch messbares Signal erfolgt bei dem betrachteten Drucksensorsystem [2] nach dem kapazitiven Wandlerprinzip. Der Drucksensor stellt eine Kondensatoranordnung dar, die aus der Sensormembrane (bewegliche Elektrode), der darunter liegenden Druckkammer und der implantierten n-Wanne (feste Elektrode) besteht. Bei der Designopti-

mierung des Drucksensors wurden zwei Kriterien berücksichtigt: die Sensitivität und die Linearität im Betriebsbereich. Optimierungsparameter sind die Kammerhöhe und –länge sowie die Membrandicke.

Der in Bild 4 dargestellte Mikroaktor [14] besteht aus zwei übereinander geschichteten dünnen Kunststofffolien, in denen mit einem Excimerlaser für die Aktorplatte, Aktorkammer und fluidischen Anschlüssen die entsprechenden Mikrostrukturen eingebracht wurden. Durch eine entsprechende Anzahl, Anordnung und Dimensionierung der ringförmigen und konzentrisch angeordneten Kerben kann das mechanische Verhalten der Aktorplatte gezielt beeinflusst werden. Ein wichtiges Optimierungsziel bei dem Kerbendesign besteht darin, bei einem gegebenen Arbeitsdruck eine möglichst große Auslenkung der Aktorplatte zu erreichen. Da die Aktorplatte gleichzeitig eine lange Lebensdauer haben soll, besteht das zweite Optimierungsziel darin, die Spannung in der Aktorplatte bei Auslenkung möglichst gering zu halten.

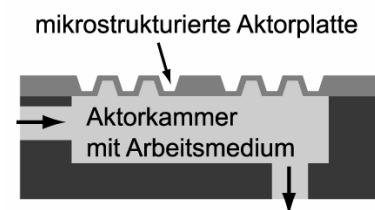


Abb.4: Schematische Darstellung der Aktorplatte

Die hier vorliegende Optimierungsaufgabe beschränkt sich somit nicht auf eine rein geometrische Optimierung, sondern sie umfasst zusätzlich die Optimierung der Topologie in Form von Anzahl und Anordnung der Kerben. Dies führt zu einem Parametervektor dynamischer Länge (pro Kerbe 4 Parameter), wodurch die meisten deterministischen und stochastischen Optimierungsverfahren nicht mehr anwendbar sind. Auch die Implementierungen vieler EAs erlauben im Gegensatz zu GLEAM keine Chromosomen dynamischer Länge, obwohl die Methode dies zulässt. Durch die diskrete Anzahl der Kerben ist der Lösungsraum zudem multimodal. Aufgrund der Genauigkeitsanforderungen und der Topologievariabilität ist der Einsatz von FEM-Modellen bei der Bewertung einzelner Lösungsvorschläge unumgänglich. Dies führt auf

einer Sun-Workstation (Ultra Sparc 10) zu Evaluationszeiten zwischen 3 und 10 Minuten. Die Optimierungen wurden daher mit einer Parallelversion von GLEAM durchgeführt [15]. Diese Ergebnisse dienen als Referenzwerte für die hier vorgestellten Resultate der hybriden Optimierung.

4 Ergebnisse

Die auf dem Statusseminar präsentierten Ergebnisse [3] und die Resultate weitergehender Untersuchungen [5] führen hinsichtlich der Hybridisierungsarten zu folgenden Schlussfolgerungen:

1. Die Vorooptimierung bringt bei einigen Testfällen eine gewisse Verbesserung, wobei einmal der Complex und ein anderes mal der Rosenbrock besser abschneidet.
2. Die Nachoptimierung verbessert zwar in der Regel die Ergebnisse der Evolution, erreicht aber lange nicht die Zuverlässigkeit bei der Lösungsfindung wie GLEAM.
3. Die (verzögerte) direkte Integration des Rosenbrock findet immer das Optimum (**GR** und **GvR**). Bei Verwendung des Complex (**GC** und **GvC**) wird meist weniger Aufwand benötigt als beim Rosenbrock, sofern die Complex-Integration als solche funktioniert.

Daher werden im folgenden nur die verzögerte und unverzögerte direkte Integration behandelt und ihre Resultate verglichen.

4.1 Ergebnisse der Testfunktionen

Bei der direkten Integration wurde Lamarck'sche Evolution verwendet, das heißt, dass der lokal verbesserte Nachkomme im Falle seiner Akzeptanz auch genotypisch an das gefundene (lokale) Optimum angepasst wurde. Bei der verzögerten direkten Integration werden die in Tab.2 angegebenen drei Parametrierungen für die Nischenbildung verglichen:

Parametrierung	e	e_{Pop}
P1	0.005	0.01
P2	0.002	0.005
P3	0.001	0.003

Tabelle 2: Steuerparameter

N_{max} variiert in Abhängigkeit von der Populationsgröße P: 2 für $P < 30$, 3 für $P = 30$, 4 für $P = 50$ und 5 für $P > 50$. Um die Ergebnisse der lokalen Verfahren mit GLEAM und den HyGLEAM vergleichen zu können, muss die Anzahl der Läufe bestimmt werden, bei der eine vergleichbare Sicherheit besteht, den Zielwert zu finden. Angesichts eines Stichprobenumfangs von 100 mag eine Sicherheit von 95,5% genügen. Damit ergeben sich die in Tab. 3 dargestellten Resultate.

Testfunktion	Genauigkeit	GLEAM	Aufwand bei 99,5% Sicherheit		HyGLEAM			
			Rosenbrock	Complex	GR	GvR	GC	GvC
Bracken	10^{-3}	2 509	139 620	305	31 264	21 834	439	545
	10^{-6}	5 000	-	392	314 834	546 499	1 433	1 785
Beale	10^{-3}	28 936	58 248	164	14 793	18 088	811	897
	10^{-6}	76 229	229 162	538	48 300	48 613	1 319	1 419
Bigg	10^{-3}	3 370 501	120 030	3 936	300 247	5 675 628	24 493	136 190
	10^{-6}	-	500 688	40 749	556 809	7 663 112	30 143	127 627
Helical Valley	10^{-6}	47 288	75 310	-	67 950	8 118	2 576	2 674
Rastrigin	10^{-4}	51 721	-	-	43 864	47 378	-	4 267 176
Fletcher&Powel	10^{-5}	483 566	55 590	12 393	13 535	10 206	4 684	4 635
Fraktale Fkt.		195 100	-	-	30 626	30 473	56 287	952 620

Tabelle 3: Vergleich der benötigten Individuen

(jeweils 100 von 100 Läufen erfolgreich oder hochgerechnet auf 95,5% Erfolgswahrscheinlichkeit)

Es fällt auf, dass der Complex in den Fällen, in denen er konvergiert, das beste Ergebnis der drei Verfahren liefert. Tab. 4 und Abb. 5 vergleichen die erreichten Verbesserungen gegenüber GLEAM: kein Eintrag bedeutet, dass die Aufgabe nicht gelöst werden konnte; eine Null bzw. ein flaches Feld, dass zur erfolgreichen Lösung mehr Aufwand notwendig war. Die Testfunktionen sind entsprechend der besseren Übersichtlichkeit im Diagramm

sortiert. Die geforderte Genauigkeit des Zielwertes ist in Klammern angegeben. Bigg (6) wurde in der Tabelle weggelassen, da ein Vergleich mit GLEAM nicht möglich ist. Für die Hybridisierung ist der Fall aber insofern ein Erfolg, als dass die direkte Complex-Integration am besten abschneidet. In Abb. 5 musste Bigg (3) aus Gründen der Darstellbarkeit weggelassen werden.

Test-Funktionen	Verbesserung gegenüber GLEAM						Parametrierungen					
	Rosen.	Compl.	GR	GvR	GC	GvC	GLEAM	Rosen.	GR	GvR	GC	GvC
Rastrigin			1,2	1,1			5		5 m	5 m P3		
Fraktale F.			6,4	6,4	0	0	20		5 n	5 n P2	10	5 P2
Bracken (3)	0	8,2	0	0	5,7	4,6	20	m	5 n	5 n P2	5	5 P2
Bracken (6)		12,8	0	0	3,5	2,8	10		5 n	5 n P2	5	5 P2
Hel.Valley	0		0	5,8	18,4	17,7	5	m	5 m	10 m P2	5	5 P2
Fletcher&P.	8,7	37,4	35,7	47,4	103,3	104,2	600	m	10 m	5 m P2	5	5 P3
Beale (3)	0	176,4	2,0	1,6	35,7	32,3	10	x	5 n	5 n P2	5	5 P2
Beale (6)	0	141,7	1,6	1,6	57,8	53,7	10	x	5 h	5 x P2	5	5 P2
Bigg (3)	28,1	856,3	11,2	0	137,6	24,7	500	h	30 m	20 m P2	20	5 P2

Tabelle 4: Verbesserung gegenüber GLEAM und beste Parametrierungen

(Parametrierungen: Populationsgröße, Rosenbrock-Abbruchschranke, Nischenparametrierung (s. Tab.2))

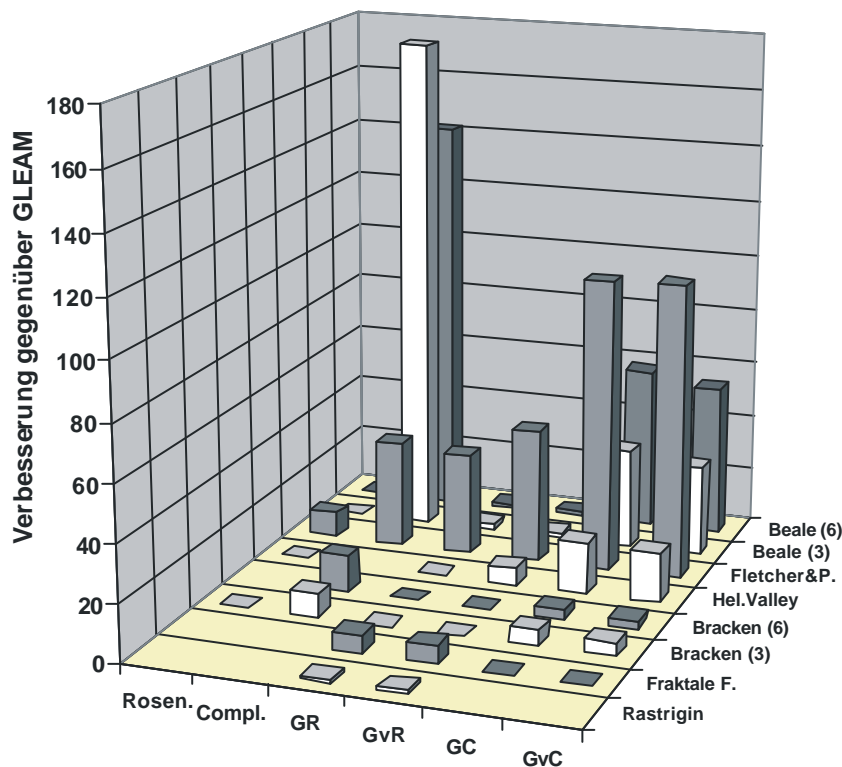


Abb.5: Vergleich der erreichten Verbesserungen des jeweils besten Laufs gegenüber GLEAM.

(kein Eintrag: Zielwert nicht erreicht, flacher Eintrag: Zielwert bei höherem Aufwand als GLEAM erreicht)

Man kann feststellen, dass die hinsichtlich Parameteranzahl und Suchraumeigenschaften einfacheren Funktionen Bracken, Beale und Bigg am besten durch den Complex gelöst werden und die breiter angelegte globale Suche von GLEAM zu aufwendig für das Problem ist oder im Falle von Bigg GLEAM ohne Unterstützung des Complex-Verfahrens kaum eine Chance hat. Bei den schwierigeren Problemen versagt bis auf Fletcher & Powel der Complex und die Hybridisierung schneidet regelmäßig am besten ab. Allerdings arbeitet GLEAM dabei mit wechselnden Partnern. Betrachtet man nun die Parametrierungen der erfolgreichsten Läufe auf der rechten Seite von Tab. 4, so kann man folgendes feststellen: Alle hybriden Varianten kommen mit kleinen Populationsgrößen zwischen 5 und 30 aus, die Abbruchschranke des Rosenbrock-Verfahrens nimmt alle Werte zwischen n (niedrig) und x (sehr hoch) an und die Nischenparametrierung P2 dominiert. Es gibt also keine allgemeingültig günstige Parametrierung. Insbesondere ist unklar, welches lokale Verfahren besser geeignet ist und im Falle des Rosenbrock, mit welcher Abbruchschranke gearbeitet werden soll. Darauf wird im letzten Abschnitt dieses Kapitels noch weiter eingegangen werden.

4.2 Ergebnisse der Designoptimierungen

Der Vergleich der verschiedenen Verfahren und Parametrierungen erfolgt auf der Basis der zur Erreichung einer bestimmten Designqualität notwendigen Evaluationen. Da die Ergebnisse bereits auf dem Statusseminar [3] ausführlich dargestellt wurden, wird hier nur eine Zusammenfassung gegeben.

4.2.1 Heterodynempfänger

Wegen des relativ hohen Simulationsaufwands wurden jeweils nur 50 Läufe durchgeführt. Die lokalen Verfahren erreichen das Qualitätsziel mit unterschiedlicher Zuverlässigkeit, was in Tab. 5 mit den sich daraus ergebenden Aufwänden für eine Sicherheit von 99,5% dargestellt ist. GLEAM erreicht ab einer Populationsgröße von 150 zuverlässig die Zielqualität, wobei der niedrigste Aufwand bei einer Populationsgröße von 180 entsteht, siehe Abb.6.

Verfahren	Rosen. Abbr.	Erfolg [%]	Aufwand [Eval.]	95,5% Sicherheit:	
				Läufe	Eval.
Rosenbrock	n	8	94	64	6 016
	m	6	232	86	19 952
	h	30	764	15	11 460
Compl.		12	102	42	4 284

Tabelle 5: Ergebnisse der lokalen Verfahren

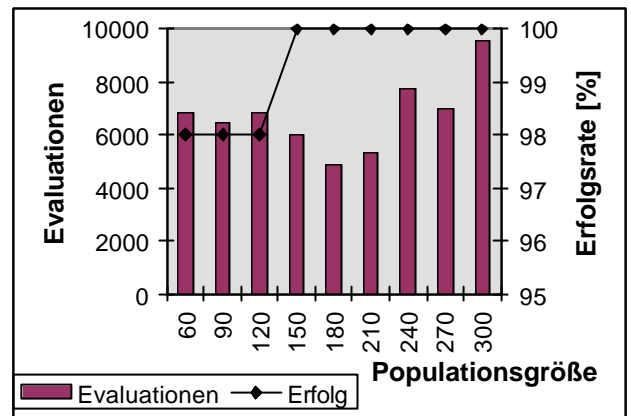


Abb.6: Aufwand und Erfolgsrate von GLEAM

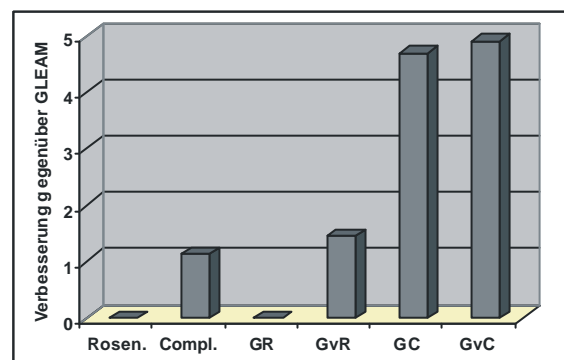


Abb.7: Vergleich der erreichten Verbesserungen des jeweils besten Laufs gegenüber GLEAM

Abb. 7 vergleicht die erreichten Verbesserungen gegenüber GLEAM basierend auf den jeweils besten Läufen der Hybriden und der berechneten Werte für die lokalen Verfahren. Alle Hybriden arbeiten mit einer Populationsgröße von 5 und die günstigste Hybridisierung GvC mit der Nischenparametrierung P3.

4.2.2 Drucksensor

Die Ergebnisse basieren auf 100 Läufen pro Parametrierung und Verfahren. Bei der hier benutzten strengeren Bewertung als in [2] erreichen die beiden lokalen Verfahren das Optimierungsziel in maximal 3% der Fälle (Rosenbrock) bzw. gar nicht (Complex). Bei

GLEAM wird erst ab einer Populationsgröße von 210 das Qualitätsziel zuverlässig erreicht und ab 360 auch bei niedrigen Evaluationsraten.

Die direkte Integration des Rosenbrock-Verfahrens mit Abbruchschranke s bringt hingegen stabile Verhältnisse, siehe Abb. 8. Der Aufwand ist mit dem der besten GLEAM-Läufe über einen weiten Bereich der Populationsgröße vergleichbar. Am besten schneidet eine Begrenzung der Iterationen des Rosenbrock auf 100 ab (Ros-100), während bei 250 (Ros-250) etwas mehr Evaluationen bei gleichem Erfolg benötigt werden. Andere Abbruchschranken führen zu deutlich schlechteren Ergebnissen. Die Integration mit dem Complex führt hier hingegen zu einem Mehraufwand.

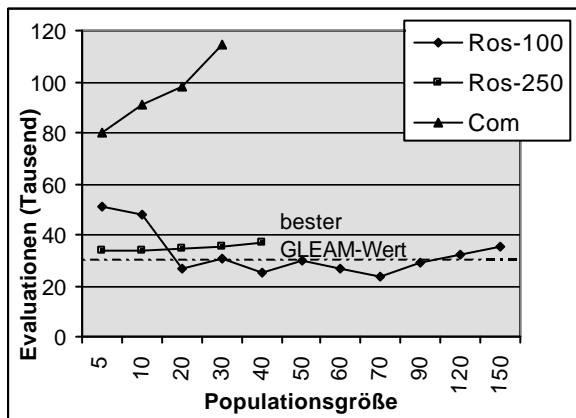


Abb.8: Direkte Integration

4.2.3 Aktorplatte

Bei der Aktorplatte verbieten sich auf Grund der langen Simulationszeiten vergleichende Läufe mit unterschiedlichen Parametrierungen. Stattdessen sind die bisher gemachten Erfahrungen praktisch anzuwenden. Verglichen werden Ergebnisse, die mit einer Parallelversion von GLEAM erzielt wurden [15], mit HyGLEAM unter Verwendung des Rosenbrock-Verfahrens bei direkter Integration und mittlerer Abbruchschranke, wobei die Hälfte der Startpopulation vorgeneriert und voroptimiert wurde [14]. Tab. 6 zeigt die Werte eines anfänglichen manuellen Designs und die der beiden Optimierungsergebnisse. Beim Vergleich des Aufwands ist folgendes zu beachten: Die Komponente zur Ankopplung des FEM-Simulators speichert die simulierten Parametervektoren zusammen mit den

Ergebnissen ab und vergleicht neue Anforderungen mit den bereits errechneten, um so Simulationszeit zu sparen. Dies kommt besonders beim Rosenbrock-Verfahren zum Tragen, da in der Konvergenzphase relativ häufig sehr ähnliche Vektoren berechnet werden, die vom Filter als identisch bewertet werden. Dementsprechend wird in der Tabelle zwischen angeforderten und ausgeführten Simulationen unterschieden. HyGLEAM benötigt also nur etwa die Hälfte an real durchgeführten Simulationen, um ein besseres Ergebnis zu erzielen als die nichthybride Variante. Dies führte zu einer Laufzeit von 2 Tagen und 15 Stunden auf 8 Suns unterschiedlicher Leistung (504 CPU-Stunden) bei GLEAM, während HyGLEAM mit 12 Tage und 16 Stunden (304 CPU-Stunden) auf einer Sun Ultra Sparc 10 auskam. Neuere Ergebnisse zu variierenden Drücken und E-Modulen sind in [16] im gleichen Band zu finden.

	Auslenkung		Spannung		Evaluierungen	
	[μm]	[%]	[N/mm^2]	[%]	angef.	ausgef.
manuell	104	100	12	100		
GADO	238	229	16	133	28978	11302
HyGADO	251	241	17	142	122847	5935

Tab.6: Optimierungsergebnisse

4.3 Schlussfolgerungen

Wie in den beiden vorigen Abschnitten dargestellt, gibt es keine problemunabhängige günstige Parametrierung für die Hybridisierung: Bei einer neuen Aufgabe ist weder klar, welches der beiden lokalen Verfahren günstiger ist, noch mit welcher Abbruchschranke der Rosenbrock betrieben werden soll oder ob und wenn ja mit welcher Verzögerung die lokale Suche zuzuschalten ist. Für die Populationsgröße kristallisiert sich immerhin heraus, dass kleine Werte genügen. Sicherheits halber sollte man einen Wert von zehn und bei wahrscheinlich schwierigen Problemen von 20 verwenden. Für die anderen Einstellungen sind selbstadaptive Lösungen anzustreben. So kann die Aufteilung zwischen Rosenbrock und Complex ausgehend von einer Gleichgewichtung durch die Bewertung

von jeweils erreichtem Erfolg und dazu notwendigem Aufwand angepasst werden.

Man kann davon ausgehen, dass bei der Nachkommensverbesserung anfänglich auf das genaue (lokale) Optimum verzichtet werden kann und die Präzision und damit der Aufwand erst im Laufe der Evolution ansteigen brauchen. Dies kann bei beiden Verfahren durch eine Begrenzung der Iterationen erreicht werden, die im Laufe der Evolution gelockert wird. Beim Rosenbrock kommt noch die Abbruchschranke als Einstellmöglichkeit hinzu. Beide Parameter können wieder durch das Verhältnis von erreichter Verbesserung bei investiertem Aufwand gesteuert werden. Als weitere Steuergröße kommt noch die Nischenbildung in Betracht, mit der der Fortschritt der Suche beurteilt werden kann.

Als Konsequenz aus den durchgeführten Untersuchungen kann gesagt werden, dass eine Hybridisierung auf der Grundlage der (verzögerten) direkten Integration bei hinreichend schwierigen Problemen eindeutig den geringsten Aufwand an Evaluationen verursacht. Dabei ist es problemabhängig, wie viel gegenüber GLEAM oder einem der lokalen Verfahren in Falle ihrer Konvergenz eingespart werden kann. Es wurden Einsparungen bis zum Faktor 104 (Fletcher & Powel) und 137 (Bigg (3)) beobachtet. Bei nicht zu schwierigen Problemen empfiehlt es sich, auch aus Gründen des geringen Aufwands erst den Complex auszuprobieren.

5 Webbasiertes HyGLEAM mit ORBIT

5.1 ORBIT

Abb. 9 gibt einen Überblick über Remote-Anwendungen mit dem in [8] ausführlicher beschriebenen ORBIT-System. Es besteht aus den in Abb. 10 dargestellten drei Komponenten Internet Web-Server, Data-Server und ORBIT-Engine, die im folgenden kurz beschrieben werden.

Der Internet Web-Server stellt die externe Schnittstelle zum ORBIT-System dar. Er verfügt über dynamische Webseiten, die je nach Anwendung automatisch generiert werden. Hierzu zählen auch HTML-Formulare, die der

Anwender zur Konfiguration der Modelle benutzt. Nach der Durchführung der jeweiligen Anwendung können auf den Seiten des Internet Web-Servers die Ergebnisseiten abgerufen werden.

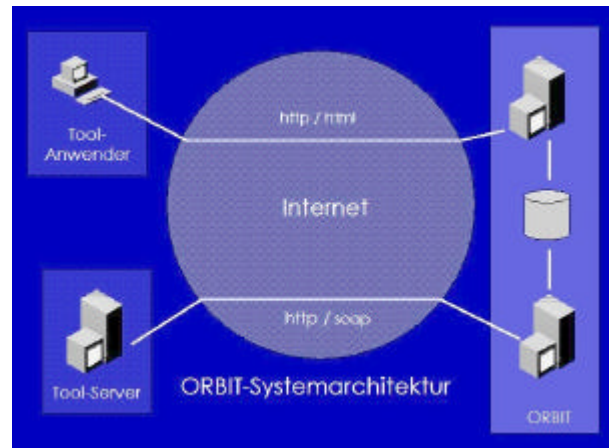


Abb.9: ORBIT-Systemarchitektur

Der Data-Server dient als Schnittstelle zwischen Internet Web-Server und ORBIT-Engine. In der Datenbank des Data-Servers werden Modell- und Ergebnisdaten sowie Laufzeitdaten von instanziierten Workflows vorgehalten. Der Data-Server verfügt über eine SOAP-Schnittstelle zur Generierung neuer Modellstrukturen und der Konfiguration von Tools bzw. Workflows.

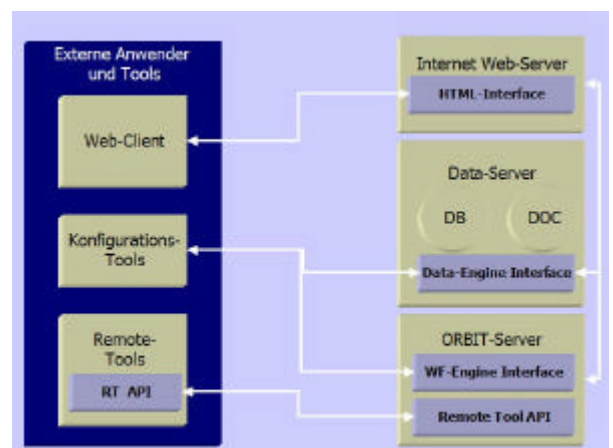


Abb.10: Komponenten des ORBIT-Systems

Die ORBIT-Engine stellt den eigentlichen Kern des Systems dar. Sie verwaltet Workflowdefinitionen und -instanzen und steuert den Fortschritt von laufenden Workflowinstanzen. Über die Remote-Tool-API werden externe Tools über SOAP in den Workflowprozess integriert und die Toolaufrufe gesteuert.

ert. Darüber hinaus verfügt die ORBIT-Engine über eine SOAP-basierte Service-Schnittstelle, mittels derer externe Tools Zugriff auf Konfigurationsdaten im ORBIT-System erhalten.

5.2 HyGLEAM in ORBIT

Zur Einbindung von HyGLEAM in das ORBIT-System wurde eine HyGLEAM-Engine ohne Benutzerinterface implementiert, deren Steuerung vollständig über die Aufrufparameter und Dateien erfolgt. Die HyGLEAM-Engine wird in einen eigenen in Java implementierten Tool-Agenten eingebettet, der die Kommunikation mit der ORBIT-Engine übernimmt. Abb. 11 zeigt das Einstiegsformular

des webbasierten Aufrufs eines Optimierungslaufs. Zuerst ist eine vorkonfigurierte Aufgabenstellung auszuwählen und dann das Optimierungsverfahren zu bestimmen. Je nach Wahl des Verfahrens gelangt man dann zu einem Unterformular, mit dessen Hilfe man die notwendige Parametrierung vornehmen kann, wobei für alle Eingabefelder Defaultwerte angeboten werden. In Abb. 12 ist das Formular für GLEAM selbst dargestellt während Abb. 13 die Parametrierung der Hybridisierungsart „Vorooptimierung“ zeigt. Während eines Laufs kann der Benutzer Zwischenergebnisse abfragen und gegebenenfalls den Lauf abbrechen. Am Ende eines Laufs erhält er die Ergebnisdaten.

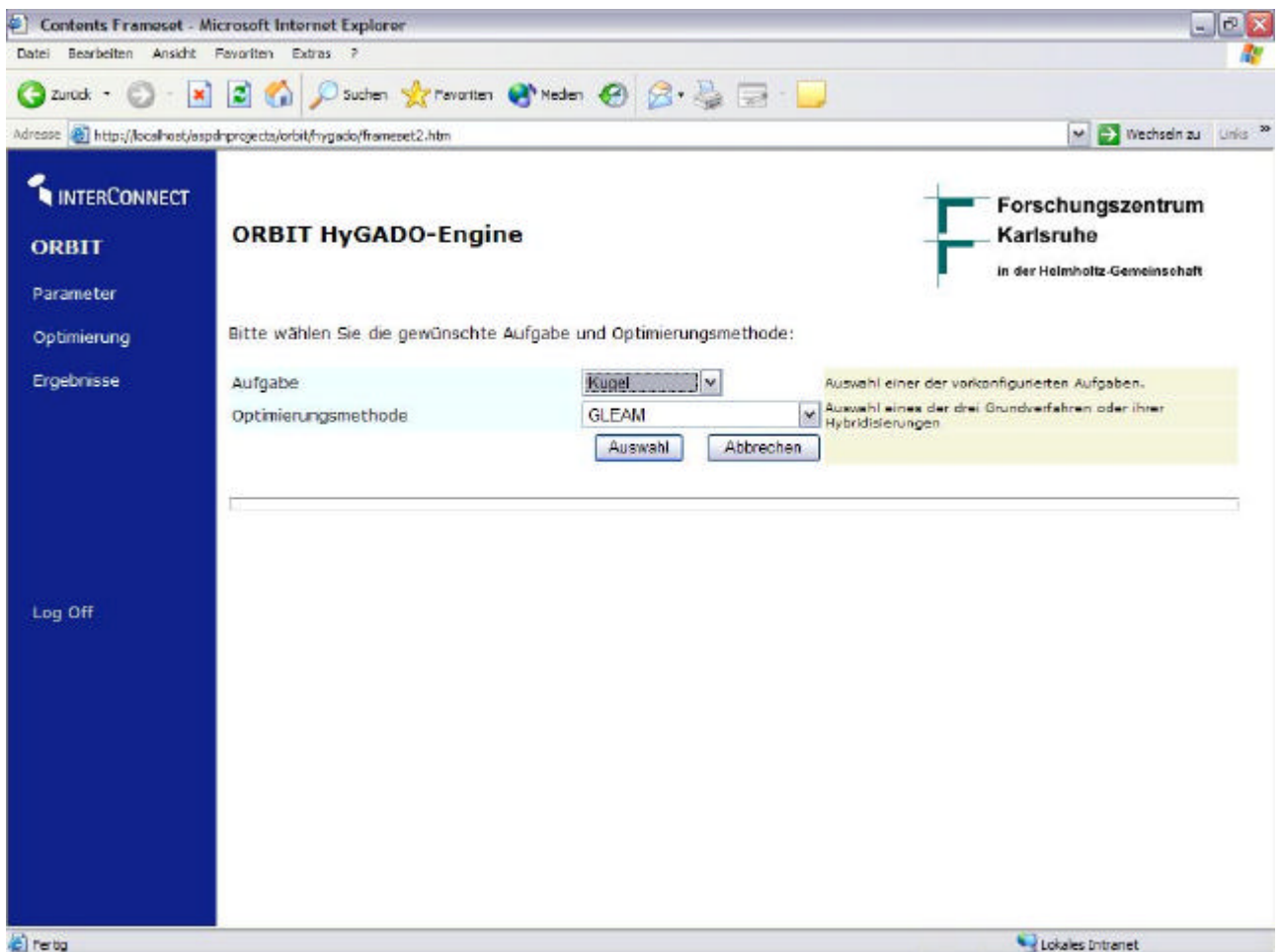


Abb.11: Einstiegsformular für einen Optimierungslauf

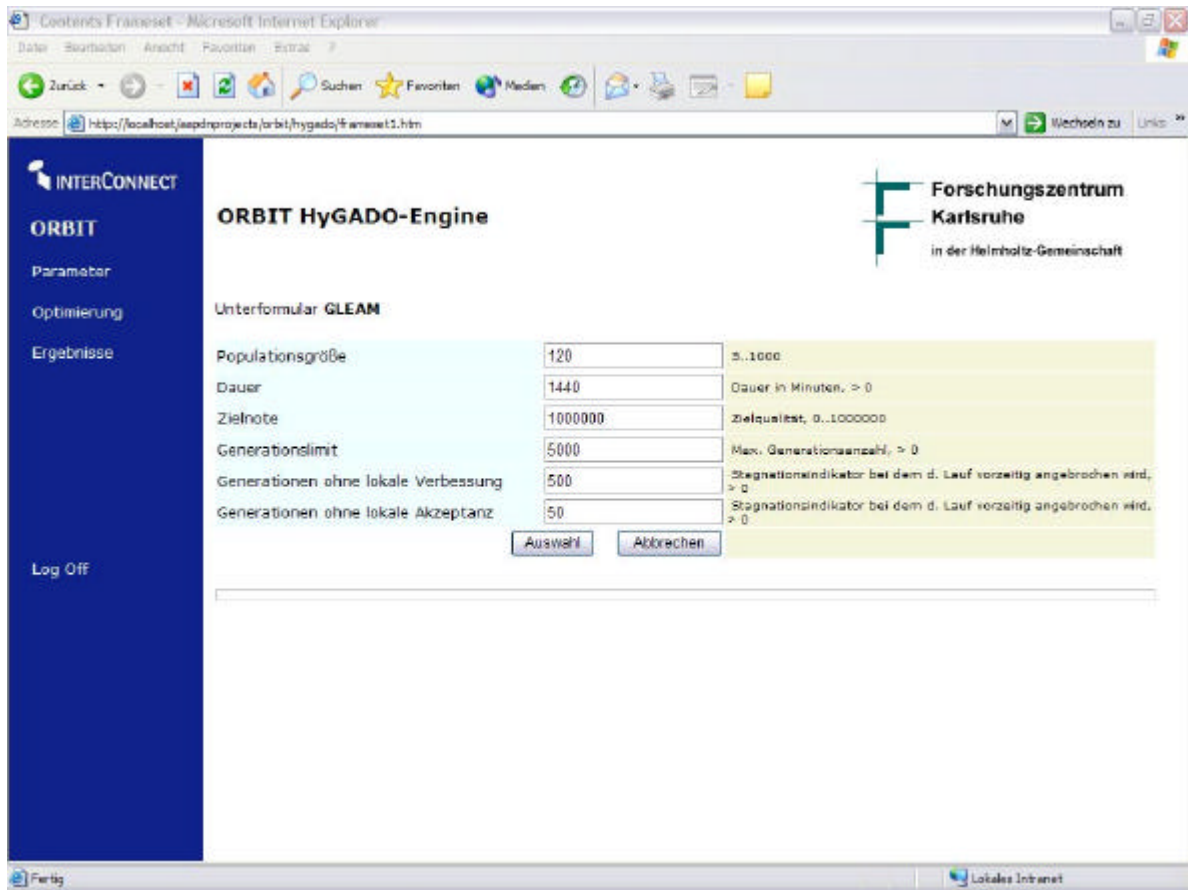


Abb.12: Unterformular zur Parametrierung eines GLEAM-Laufs

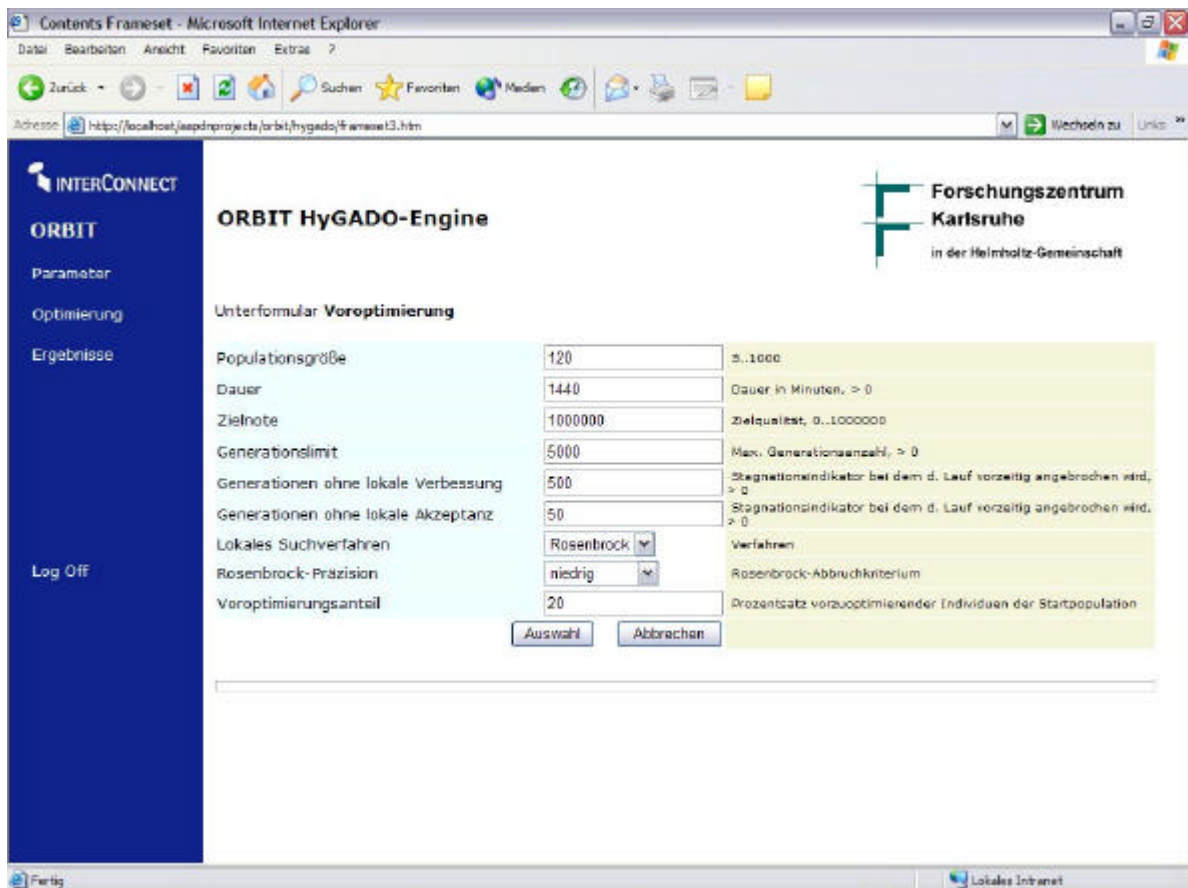


Abb.13: Unterformular zur Parametrierung der Hybridisierungsart „Voroptimierung“

6 Zusammenfassung

Die wichtigsten Ergebnisse der umfangreichen Untersuchungen zum Verhalten globaler und lokaler Suchverfahren basierend auf dem Evolutionären Algorithmus GLEAM, dem Rosenbrock-Algorithmus und dem Complex-Verfahren unter Verwendung mathematischer Benchmarkfunktionen und realer Designprobleme sind, dass

1. es kein generell günstigstes Verfahren gibt,
2. bei einfacheren Aufgaben mit geringer Parameteranzahl die lokale Verfahren mit weniger Aufwand zum Ziel führen als das global arbeitende GLEAM,
3. ab einem gewissen Schwierigkeitsgrad die Hybridisierung in Form der direkten Integration zuverlässig und mit vergleichsweise geringem Aufwand zum Ziel führt,
4. hinsichtlich einer adaptiven Steuerung der Rechenzeitaufteilung zwischen globaler und lokaler Suche und der adaptiven Auswahl des geeigneten lokalen Verfahrens weiterer Forschungs- und Entwicklungsbedarf besteht, siehe Abschn. 4.3.

Mit der webbasierten Version von HyGLEAM wird eine Remote-Optimierung und damit eine wirtschaftlichere Nutzung der Simulationswerkzeuge durch einen Application Service Provider ermöglicht.

Literatur

- [1] D. Peters, M. Gorges-Schleuter, W. Jakob, S. Parodat: *Modellbasierte Optimierung mit mathematischen Methoden – ein Vergleich von klassischen und evolutionären Verfahren*. 7. GMM-Workshop Methoden und Werkzeuge zum Entwurf von Mikrosystemen. Paderborn, 1999.
- [2] W. Jakob, B. Knorr, S. Parodat, D. Peters, A. Uhlig: *Optimierung von Mikrosystemen*. 8. GMM-Workshop Methoden und Werkzeuge zum Entwurf von Mikrosystemen. Berlin, 1999.
- [3] W. Jakob, D. Peters: *Designoptimierung mit HyGADO*. In: R. Laur, D. Peters (Hrsg.): OMID – Optimierung von Mikrosystemen für Diagnose- und Überwachungsanwendungen (Statusseminar), VDI/VDE, 2001.
- [4] W. Jakob: *HyGLEAM: Hybrid General purpose Evolutionary Algorithm and Method*. Proc. of 5th World Multiconference on Systemics, Cybernetics and Informatics (SCI'2001), 2001.
- [5] W. Jakob: *HyGLEAM - An Approach to Generally Applicable Hybridization of Evolutionary Algorithms*. Proc. of PPSN VII, LNCS 2439, Springer-Verlag, Berlin, 2002.
- [6] C. Blume: *GLEAM - A System for Simulated Intuitive Learning*. Proc. of PPSN I, LNCS 496, Springer-Verlag, 1991.
- [7] C. Blume, W. Jakob: *GLEAM - an Evolutionary Algorithm for Planning and Control Based on Evolution Strategy*. In: E. Cantú-Paz (Hrsg.): GECCO 2002, Vol. Late-Breaking Papers, L. Livermor National Laboratory, 2002.
- [8] A. Reiffer, P. Rohnacher: ORBIT – Internetbasierte Plattform für Remote-Anwendungen. In: H. Eggert, A. Reiffer (Hrsg.): OMID – Optimierung von Mikrosystemen für Diagnose- und Überwachungsanwendungen (Abschlußseminar des BMBF-Verbundprojekts OMID), FZK, 2002.
- [9] H.-P. Schwefel: *Evolution and Optimum Seeking*. John Wiley & Sons, Chichester, 1995.
- [10] C. Blume, W. Jakob: *Cutting Down Production Costs by a New Optimization Method*. Proc. of the Japan - U.S.A. Symposium on Flexible Automation, ASME, 1994
- [11] H. H. Rosenbrock: *An Automatic Method for Finding the Greatest or Least Value of a Function*, Comp. J. 3, 1960.
- [12] M. J. Box: *A New Method of Constrained Optimization and a Comparison with other Methods*, Comp. J. 8, 1965.
- [13] I. Sieber, H. Eggert, H. Guth, W. Jakob, K.-P. Scherer, P. Ziegler: *Design Optimization Considering Tolerance Effects of Microoptical Benches*. MicroSystem Technologies 98, VDE-Verlag, 1998.
- [14] A. Quinte: *Modellierung und Optimierung der Aktorplatte*. In: R. Laur, D. Peters (Hrsg.): OMID – Optimierung von Mikrosystemen für Diagnose- und Überwachungsanwendungen (Statusseminar), VDI/VDE, 2001.
- [15] W. Jakob, A. Quinte, K.-P. Scherer, H. Eggert: *Optimization of a Micro Fluidic Component using a Parallel Evolutionary Algorithm and Simulation Based on Discrete Element Methods*. S. Hernández, C.A. Brebbia (Eds.): Computer Aided Optimum Design of Structures VII, WIT Press, Southampton, 2001.
- [16] A. Quinte: *Modellierung und Optimierung der Aktorplatte*. In: H. Eggert, A. Reiffer (Hrsg.): OMID – Optimierung von Mikrosystemen für Diagnose- und Überwachungsanwendungen (Abschlußseminar des BMBF-Verbundprojekts OMID), FZK, 2002.