

DESIGNOPTIMIERUNG MIT HyGADO

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

Kurzfassung

Durch die Integration allgemein anwendbarer lokaler Suchverfahren in globale Verfahren wie die Evolutionären Algorithmen sollen deren Vorteile unter Vermeidung ihrer jeweiligen Nachteile kombiniert werden. Dazu wird eine neue Methode zur Steuerung des Verfahrenseinsatzes vorgestellt. Die Robustheit und die gesteigerte Performance des resultierenden hybriden Verfahrens wird an Hand dreier Designanwendungen nachgewiesen.

1 Einleitung

Im Verbundprojekt OMID (Optimierung von Mikrosystemen für Diagnose- und Überwachungsanwendungen) werden unter anderem die im Vorgängerprojekt DEMIS entwickelten und getesteten Optimierungsverfahren integriert und an Hand der Anforderungen neuer Aufgaben weiterentwickelt. 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 darstellt. 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. Da die Ermittlung der Qualität einer Lösung bei Designoptimierungsaufgaben in der Regel mit Hilfe von Simulationsmodellen durchgeführt wird, ist eine Verringerung der meist aufwendigen Simulationen notwendig. Die Idee liegt nahe, zu versuchen, die beiden Algorithmenklassen so geschickt miteinander zu verbinden, dass der

resultierende Hybrid die Vorteile der beteiligten Verfahren ohne deren Nachteile in sich vereint.

In Kap. 2 und 4 werden die am FZK entwickelten Hybridisierungsmethoden [3] vorgestellt. Integriert wurden zwei am ITEM implementierte lokale Suchverfahren [1] und das GLEAM-Verfahren (General Learning Evolutionary Algorithm and Method) [4], das am FZK zu GADO (Genetischer Algorithmus zur Designoptimierung) weiterentwickelt wurde [5], siehe Kap.3. Erprobt und getestet wurde das resultierende hybride GADO (HyGADO) an Hand dreier in Kapitel 5 kurz vorgestellter Anwendungen. Die Ergebnisse sind in Kapitel 6 zu finden.

2 Arten der Hybridisierung

Folgende Arten der Hybridisierung wurden untersucht:

- 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 meist zulässigen Lösungen einer gewissen Qualität beginnen zu lassen und so die Zeit zum Auffinden befriedigender Bereiche des Suchraums zu sparen.
- Nachoptimierung der GADO-Ergebnisse
Die Nachoptimierung der GADO-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 in Kap. 4 eingegangen.

- Direkte Integration
Hierbei werden alle oder nur der beste Nachkomme einer Paarungsoperation von GADO mit dem lokalen Verfahren optimiert. Damit operiert GADO 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 dazuschalten. Die Frage, ab wann dies geschehen soll, wird ebenfalls in Kap. 4 behandelt.

3 Benutzte Optimierungsverfahren

3.1 GADO und GLEAM

Das in GADO enthaltene GLEAM ist ein eigenständiger evolutionärer Algorithmus, der Elemente der (reelcodierten) Genetischen Algorithmen und der Evolutionsstrategie 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 Aktionskette (AK), die den Chromosomen entspricht. Es gibt (derzeit) in GLEAM 3 Chromosomen- oder AK-Typen:

- AK fester Länge, irrelevante Reihenfolge
Jeder Aktionstyp ist mit genau einer Aktion vertreten.
- AK fester Länge, relevante Reihenfolge
Jeder Aktionstyp ist mit genau einer Aktion vertreten.
- AK variabler Länge, relevante Reihenfolge
Jeder Aktionstyp kann mit keiner, einer oder mehreren Aktionen vertreten sein.

Der erste Aktionskettentyp wird bei reiner Parameteroptimierung benutzt, der zweite kommt z.B. bei Scheduling-Aufgaben zum Einsatz [6], während der dritte bei anspruchsvoller Designoptimierung wie z.B. der in Abschnitt 5.3 beschriebenen Akteurplatte Verwendung findet.

GLEAM enthält einen Satz vordefinierter genetischer Operatoren (Mutation der Werte oder der Aktionsposition, verschiedene Crossover-Operatoren), die aufgabenspezifisch konfiguriert zur Anwendung gebracht werden können. Die Werte-Mutationen berücksichtigen die in den Aktionstypdefinitionen mit hinterlegten Wertebereichsgrenzen, sodass explizite Restriktionen immer eingehalten werden. Kleine Änderungen sind dabei entsprechend der Evolutionsstrategie wahrscheinlicher als große. Die Partnerwahl erfolgt ranking-basiert und pro Paarung werden mehrere Nachkommen erzeugt, wobei die dabei beteiligten genetischen Operatoren sowie ihre Anwendungswahrscheinlichkeiten problemsspezifisch ausgewählt und konfiguriert werden können. Wegen der elitären Form der Akzeptanzregel für erzeugte Nachkommen ist eine Verschlechterung der pro Generation erreichten Lösungsqualität nicht möglich.

Im Gegensatz zu vielen Evolutionären Algorithmen, bei denen sich ein Elter den Partner aus der Gesamtheit der Individuen aussuchen kann (panmiktische Population), benutzt GLEAM eine strukturierte Population mit lokaler Selektion [7]. Dabei kann man sich die Individuen auf einen Ring angeordnet vorstellen, wobei jedes Individuum eine bestimmte Anzahl Nachbarn zur rechten und zur linken hat, siehe Bild 1. Die Partnerwahl zur Nachkommensbildung und die Akzeptanz er-

zeugter Kinder findet ausschließlich in diesen Nachbarschaften (engl. deme) statt. Da sich die Nachbarschaften benachbarter Individuen mehr oder weniger überlappen, findet ein Informationsaustausch zwischen den Demes statt. Bild 1 zeigt je zwei Demes mit vier Nachbarn bei minimaler (oben) und maximaler (unten) Überlappung. Mit diesem Konzept wird einer vorzeitigen Konvergenz auf ein Suboptimum entgegengewirkt und die Bildung unterschiedlicher Nischen mit jeweils ähnlichen Lösungen guter Qualität gefördert. Dieser Ansatz mit seiner adaptiven Balance zwischen Breiten- und Tiefensuche hat sich vor allem bei multimodalen Aufgabenstellungen bewährt [5,6,8].

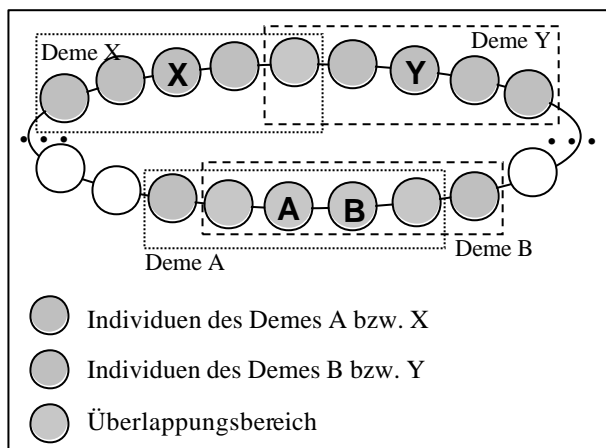


Abb.1: Beispiele für sich überlappende Demes

3.2 Lokale Suchverfahren

Zur Entwicklung von HyGADO (bzw. HyGLEAM) wurden zwei deterministische, ableitungsfrei arbeitende lokale Suchverfahren herangezogen: der Complex-Algorithmus nach Box [9,10,11] und das rotierende Koordinatenverfahren nach Rosenbrock [9,10,12]. Beide können wie GLEAM Aufgabenstellungen mit den bei praktischen Problemen üblicherweise auftretenden Restriktionen behandeln. HyGADO liefert in der Regel mehrere unterschiedliche Ergebnisse vergleichbarer Qualität, die zur Bildung des Startcomplexes benutzt werden können. Damit gibt es beim Complex zwei Varianten der Nachoptimierung: Die Verwendung aller HyGADO-Ergebnisse als separate Startpunkte entsprechend vieler Complex-Läufe oder zur Bildung eines einzigen Startcomplexes.

4 Steuerung von evolutionärer und lokaler Suche

Das Umschalten von evolutionärer auf lokale Suche oder das Hinzuschalten der lokalen bei verzögerter direkter Integration ist sicher dann sinnvoll, wenn die Population zu wenigen einander jeweils ähnlichen Subpopulationen konvergiert ist. Dazu ist die Bestimmung des genotypischen Abstands der Individuen notwendig. Da diese Berechnungen aufwendig sind, sollten sie nur bei hinreichender Aussicht auf Erfolg durchgeführt werden. Ein geeigneter Auslöser für diese Prüfung ist das Auftreten einer gewissen Anzahl von Generationen ohne Verbesserung der bisher erreichten Qualität. Dieser bei jedem EA mögliche Trigger kann bei GADO mit seinen Nachbarschaften insofern verfeinert werden, als hier die Messung der Generationen ohne Verbesserung innerhalb aller Demes (GDV) oder, noch strenger, ohne Akzeptanz eines Nachkommens in allen Demes (GAK) herangezogen werden.

4.1 Abstandsmaße

Zur Bestimmung des genotypischen Abstands muss zwischen den verschiedenen Chromosomentypen von GLEAM unterschieden werden. Die nachstehend vorgestellten Abstandsmaße zweier nichtleerer Chromosomen wurden so gewählt, dass sie sich möglichst einfach berechnen lassen und unabhängig von den Wertebereichsgrenzen und anderen Definitionen der jeweiligen Aktionstypen sind. Außerdem wurde ihr Wertebereich auf das Intervall 0 (kein Abstand) und 1 (maximaler Abstand) begrenzt. Die Abstandsmaße entsprechen den Anforderungen an eine Metrik. Die zugehörigen Beweise und die Herleitung von $dist_{max}$ wurden hier aus Platzgründen weggelassen und sind unter folgender Web-Adresse zu finden:

http://www.iai.fzk.de/~jakob/hy_gleam/

4.1.1 Chromosome fixer Länge mit irrelevanter Aktionsreihenfolge

Bei diesem einfachsten der drei Chromosomentypen genügt die Bestimmung des Parameterabstands Δ_{par} der beiden Chromosomen C_1 und C_2 :

$$\Delta_{par}(C_1, C_2) = \frac{1}{n} \sum_{i=1}^n \frac{|par_{i,1} - par_{i,2}|}{og_i - ug_i}$$

- mit n Anzahl der Parameter aller Aktionen
 $par_{i,j}$ i -ter Parameter von Chromosom j
 ug_i Untere Grenze des Wertebereichs des i -ten Parameters
 og_i Obere Grenze des Wertebereichs des i -ten Parameters

4.1.2 Chromosome fixer Länge mit relevanter Aktionsreihenfolge

Der Positionsabstand $pa_{1,2}(A_i)$ einer Aktion A_i in den beiden Chromsomen C_1 und C_2 wird dadurch definiert, dass die Aktionen jeder Kette sequentiell durchnummeriert werden und dann der Betrag der Differenz ihrer Indizes bestimmt wird:

$$pa_{1,2}(A_i) = |Idx_1(A_i) - Idx_2(A_i)|$$

mit $Idx_j(A_i)$ Index der Aktion i im Chromosom j .

Der gesamte Positionsabstand zweier Ketten, deren Länge größer als 1 ist, $\Delta_{pos}(C_1, C_2)$ wird dann definiert als:

$$\Delta_{pos}(C_1, C_2) = \frac{1}{dist_{max}} \sum_{i=1}^{len} pa_{1,2}(A_i)$$

mit len Anzahl der Aktionen im Chromosom.

Der maximale Abstand $dist_{max}$ ergibt sich, wenn die Aktionen um einen maximalen Abstand verschoben wurden. Er berechnet sich für Chromosome gerader und ungerader Länge getrennt wie folgt:

$$dist_{max,gerade} = \frac{len^2}{2} \quad dist_{max,ungerade} = \frac{len^2 - 1}{2}$$

Der Gesamtabstand $\Delta(C_1, C_2)$ ist der Durchschnitt aus dem Parameter- und dem Positionsabstand.

4.1.3 Chromosome variabler Länge mit relevanter Aktionsreihenfolge

Bei diesem Chromosomentyp spielt neben dem Parameter- und dem Positionsabstand noch der Unterschied der Aktionspräsenz in beiden Ketten eine Rolle. Letzteres beschreibt den Unterschied im Vorhandensein der Aktionen in beiden Ketten. Da zur Bestimmung des Konvergenzgrades einer Population die exakte Ermittlung geringer Unterschiede zwischen ähnlichen Chromosomen wesentlich wichtiger ist als der genaue Unterschied unähnlicher Ketten, kann man sich bei der Aufstellung der

Formeln auf die exakte Berechnung sich ähnlicher Chromosomen konzentrieren.

C_1 sei die längere der beiden Ketten und A_{gem} die Menge ihrer gemeinsamen Aktionen. Da Aktionen bei diesem Kettentyp mehrfach vorkommen können, wird festgelegt, dass sie in der Reihenfolge ihrer Indizierung betrachtet werden. Wenn A_{gem} leer ist, wird der Gesamtabstand $\Delta(C_1, C_2)$ auf 1 gesetzt.

Der Parameterabstand Δ_{par} ist über A_{gem} definiert. Dies reicht aus, da die dabei nicht bewerteten Aktionen in den noch zu definierenden Unterschied der Aktionspräsenz eingehen. Der Positionsabstand Δ_{pos} wird ebenfalls über A_{gem} definiert, wobei das $dist_{max}$ von der kürzeren Kette C_2 genommen wird. Der dabei gemachte Fehler kann vor allem bei Chromosomen mit erheblicher Längendifferenz zu einem zu groß bewertetem Δ_{pos} führen, das dabei auch größer als 1 werden kann. Da sich solche Ketten aber erheblich unterscheiden, wird dies in Kauf genommen und Δ_{pos} auf 1 begrenzt.

Der Unterschied der Aktionspräsenz wird folgendermaßen bestimmt:

$$\Delta_{akt}(C_1, C_2) = 1 - \frac{card(A_{gem})}{\max(len(C_1), len(C_2))}$$

mit $len(C_i)$ Länge des Chromosoms C_i .

$card(A)$ Anzahl der Elemente der Menge A .

Bei der Ermittlung des Gesamtabstands $\Delta(C_1, C_2)$ als Durchschnitt der drei Einzelabstände wird Δ_{akt} dreifach gewichtet, da Δ_{par} und Δ_{pos} nur über der Menge der gemeinsamen Aktionen beider Ketten definiert sind.

4.2 Steuerungskriterium

Zur Bestimmung der genotypischen Diversität einer Population werden die Individuen der Reihe nach paarweise mit Hilfe des Gesamtabstands $\Delta(C_1, C_2)$ verglichen. Zwei Individuen gelten dabei als ähnlich, wenn ihr Gesamtabstand kleiner als ein vorgegebenes ϵ ist. Wenn mehrere ähnliche Ketten hintereinander auftreten, werden die Abweichungen aufsummiert und wenn die Summe 2ϵ überschreitet, erfolgt ein Vergleich zwischen dem ersten Individuum der Sequenz und dem aktuellen. Die derart ermittelte Sequenz sich hinreichend ähnlicher Individuen wird auch als Nische bezeichnet. Einzelindividuen, deren Fitness klei-

ner als 50% der bisher erreichten besten Fitness ist, werden bei der Nischenermittlung nicht berücksichtigt, da sie zur Aufteilung der Gesamtpopulation in etablierte Nischen hoher Qualität nichts beitragen. Das mittlere Individuum einer Nische wird als ihr Repräsentant bezeichnet und dient zum Vergleich nicht benachbarter Nischen. Diese werden zusammengefasst, wenn ihre Repräsentanten einander ähnlich sind. Schließlich erhält man die Anzahl unterschiedlicher Nischen N_{Anz} und deren qualitative Unterschiede, gemessen am $\Delta(C_1, C_2)$ ihrer Repräsentanten. Das Maximum dieser Unterschiede $\Delta_{N,max}$ bildet zusammen mit der Nischenanzahl das Kriterium zur Beurteilung der Konvergenz einer Population. Es ist erfüllt, wenn

$$\Delta_{N,max} \leq e_{Pop} \quad \text{und} \quad N_{Anz} \leq N_{max}$$

In diesem Fall wird entweder die evolutionäre Suche beendet und mit Hilfe eines der lokalen Verfahren nachoptimiert oder es erfolgt die Zuschaltung der lokalen Verbesserung der Nachkommen (verzögerte direkte Integration).

Damit ergeben sich folgende externe Strategieparameter für Nischenbildung und Ablaufsteuerung:

- **GDV und GAK**
Die Generationen ohne Deme-Verbesserung bzw. ohne Akzeptanz dienen lediglich als Auslöser für die Nischenüberprüfung. Bei EAs ohne strukturierter Population können statt den GDV auch die Generationen ohne Verbesserung des global besten genommen werden.
- **e**
Grenze für den genotypischen Unterschied zweier Individuen.
- **e_{Pop}**
Grenze für das Maximum der genotypischen Unterschiede der Nischen einer Population $\Delta_{N,max}$.
- **N_{max}**
Grenze für die Nischenanzahl einer Population N_{Anz} .

5 Designoptimierungsaufgaben

Die Designoptimierungsaufgaben werden hier nur kurz vorgestellt. Eine ausführliche Be-

schreibung des Drucksensors und des Heterodynempfängers ist in [2,13] zu finden, während die Aktorplatte in einem eigenen Beitrag in diesem Band beschrieben wird [14]. Alle drei Aufgabenstellungen sind multimodal.

5.1 Drucksensor

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

5.2 Heterodynempfänger

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.

5.3 Aktorplatte

Der in Bild 2 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 Optimie-

rungsziel 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.

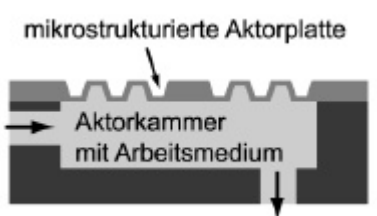


Abb.2: Schematische Darstellung der Aktorplatte

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. Erste Optimierungen wurden daher mit einer Parallelversion von GADO durchgeführt [15]. Diese Ergebnisse dienen als Referenzwerte für die hier vorgestellten Resultate der hybriden Optimierung.

6 Ergebnisse

Der Vergleich der verschiedenen Verfahren und Parametrierungen erfolgt auf der Basis der zur Erreichung einer bestimmten Designqualität notwendigen Evaluationen. Wenn bei manchen Läufen auf Grund der Überschreitung eines Maximalwertes erlaubter Evaluati-

onen vorher abgebrochen wurde, wird dies als Erfolgsrate angegeben. Fehlt sie, haben alle Läufe einer Parametrierung das Optimierungsziel erreicht.

Bei GADO bzw. HyGADO wird die Populationsgröße variiert während beim Rosenbrock unterschiedliche Abbruchschranken verglichen werden. Kleine Populationsgrößen führen zu hohen Evaluationswerten und bergen die Gefahr in sich, das Optimierungsziel nicht zu erreichen. Zu große Populationen verschwenden Rechenzeit. Kleine Abbruchschranken können ebenfalls dazuführen, dass das Optimierungsziel nicht erreicht wird, während zu große Nichtkonvergenz bewirken können. Die Abbruchschranken werden mit niedrig, mittel, hoch und sehr hoch (x) bezeichnet. Beim Complex wurde hier keine extern einstellbarer Strategieparameter benutzt. Für die Nachoptimierung und die verzögerte direkte Integration werden 3 Parametrierungen verglichen:

Tab.1: Steuerparameter

Parametrierung	ϵ	ϵ_{Pop}
1	0.005	0.01
2	0.002	0.005
3	0.001	0.003

N_{max} variiert in Abhängigkeit von der Populationsgröße P : 2 für $P < 30$, 3 für $P = 30$, 4 für $P = 50$ und 5 für $P > 50$.

Pro Parametrierung wurden 100 (Drucksensor) bzw. 50 Läufe (Heterodynempfänger) durchgeführt.

6.1 Drucksensor

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). Auch GADO hat bei kleinen Populationen Schwierigkeiten wie Bild 3 zeigt. Erst ab einer Populationsgröße P von 210 wird das Qualitätsziel zuverlässig erreicht und ab $P = 360$ auch bei niedrigen Evaluationsraten.

Die direkte Integration des Rosenbrock-Verfahrens mit Abbruchschranke s bringt hingegen stabile Verhältnisse, siehe Bild 4. Der Aufwand ist mit dem der besten GADO-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. Auch der Complex bringt eine Verbesserung, die allerdings deutlich geringer ausfällt.

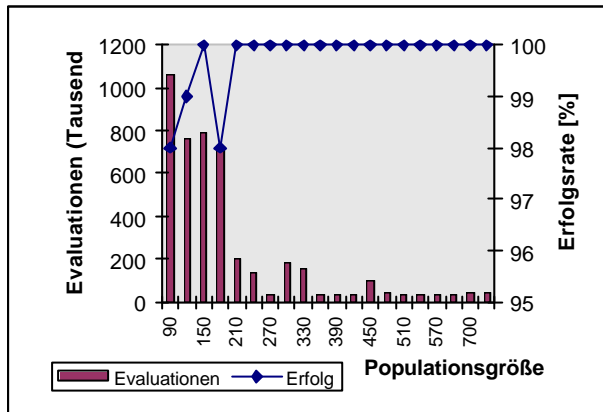


Abb.3: GADO-Ergebnisse

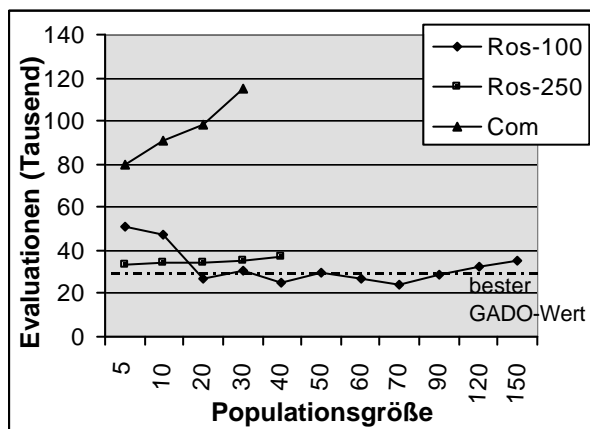


Abb.4: HyGADO (direkte Integration)

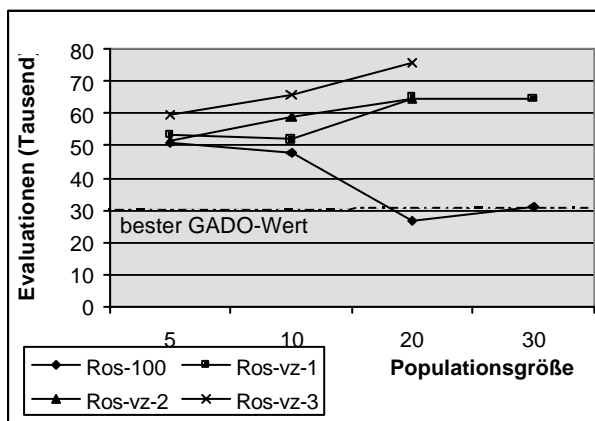


Abb.5: HyGADO (verzög. direkte Integr.)

Die in Bild 5 dargestellte verzögerte direkte Integration bringt im Vergleich zum unver-

zögerten Ros-100-Lauf keine Verbesserung. Die Nachoptimierung ist wie die Initialisierung der Startpopulation wenig erfolgreich. Die Nachoptimierung erreicht bei kleinen Populationsgrößen bei allen Parametrierungen das Qualitätsziel schlecht (zwischen 36% und 68%), während es bei größeren zwar zuverlässiger erreicht wird, allerdings in hohem Maße bereits von der Evolution, so dass es zu keiner Nachoptimierung mehr kommt. Bei mit dem Rosenbrock voroptimierten Startpopulationen wird zum Teil zuverlässig das Qualitätsziel bei geringerem Aufwand erreicht, aber es ist vorher schlecht abzuschätzen, welcher Anteil der Startpopulation günstigerweise voroptimiert werden soll. Bild 6 vergleicht nur Läufe, die zu 100% das Qualitätsziel erreichten. Mit dem Complex voroptimierte Startpopulationen bringen keine Verbesserung.

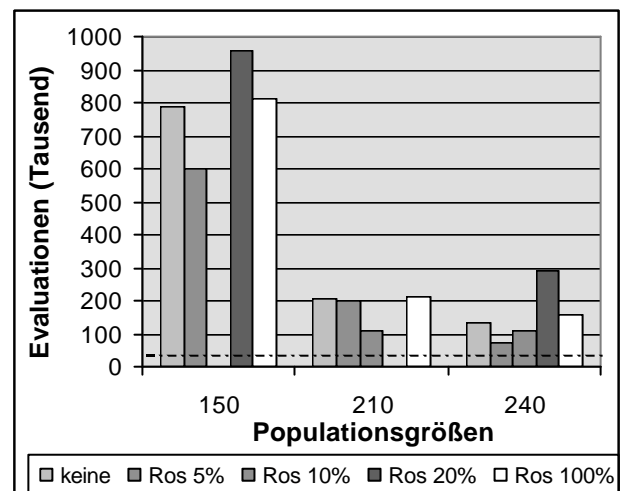


Abb.6: HyGADO (voropt. Startpopulation)

Zusammenfassend kann gesagt werden, dass die direkte Integration beim Drucksensor den größten Erfolg gebracht hat. Dieser besteht weniger in einer Verringerung des Rechenaufwands gegenüber der besten GADO-Parametrierung als in einer großen Unempfindlichkeit gegenüber der Populationsgröße: zwischen $P=20$ und $P=120$ können gute Ergebnisse bei vergleichbar geringem Aufwand erwartet werden.

6.2 Heterodynempfänger

Die lokalen Verfahren erreichen das Qualitätsziel mit unterschiedlicher Zuverlässigkeit, wie Tabelle 2 zeigt.

Tab.2: Ergebnisse der lokalen Verfahren

	Erfolg [%]	Evaluationen
Ros, Schranke n	8	94
Ros, Schranke m	6	232
Ros, Schranke h	30	764
Compl	12	102

GADO liefert ab einer Populationsgröße von 150 zuverlässig die Zielqualität, wobei der niedrigste Aufwand bei P=180 entsteht, siehe Bild 7.

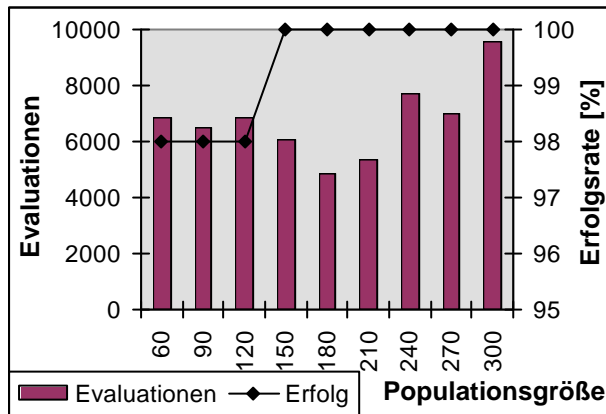


Abb.7: GADO-Ergebnisse

Bild 8 zeigt, dass die direkte Integration nur beim Complex einen Vorteil gegenüber der besten GADO-Parametrierung bringt.

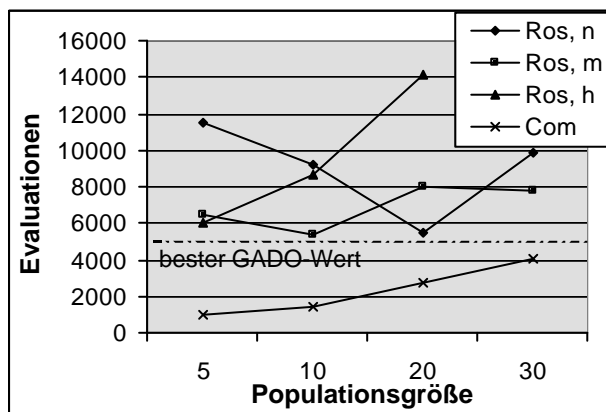


Abb.8: HyGADO (direkte Integration)

Wenn man nun noch die direkte Integration verzögert, unterbietet auch der Rosenbrock den geringsten GADO-Aufwand, wie in Bild 9 dargestellt.

Bei der Nachoptimierung ergibt sich ein ähnliches Bild wie beim Drucksensor: Bei kleinen Populationsgrößen ist die Erfolgsrate zu gering (62-84%) und bei größeren wird der Erfolg im wesentlichen ohne Nachoptimierung durch die Evolution allein erreicht. Ge-

nerell bringt die Verwendung aller GADO-Ergebnisse zur Bildung eines Startcomplexes bessere Ergebnisse als verschiedene Complex-Läufe mit je einem GADO-Resultat oder die Nachoptimierung mit dem Rosenbrock.

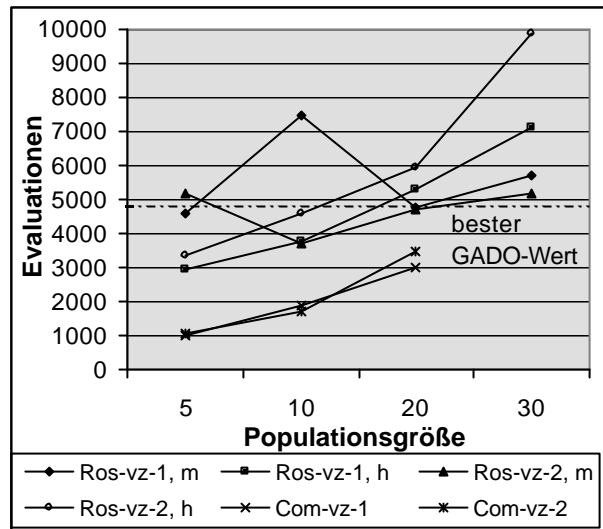


Abb.9: HyGADO (verzög. direkte Integr.)

Voroptimierte Startpopulationen bringen nur beim Complex konkurrenzfähige Ergebnisse (Bild 10). Bei vollständiger Initialisierung der Startpopulation wird die Lösung bereits in der Initialisierungsphase gefunden, ab 40 Complexläufen wird die Zielqualität also sicher erreicht, allerdings zu einem höheren Aufwand als beim besten GADO-Lauf. Initialisiert man aber nur zwischen 10 und 20% der Startpopulation gelingt eine Unterbietung des geringsten GADO-Aufwands.

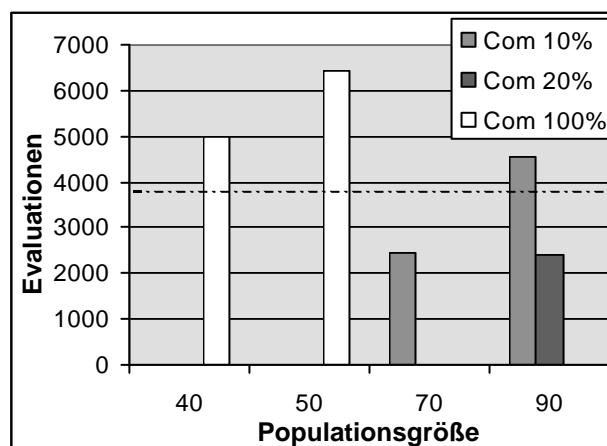


Abb.10: HyGADO (voroptimierte Startpopulation)

Bei einer Mehrfachanwendung der lokalen Verfahren kann beim Rosenbrock bei Abbruchschranke *h* ab vierfacher und beim Com-

plex ab zehnfacher Anwendung erwartet werden, dass das Qualitätsziel zuverlässig erreicht wird. Der damit verbundene Aufwand von etwa 3000 bzw. 1000 Evaluationen liegt nicht unter den besten hybriden Ergebnissen. Diese Fragestellung wurde aber nicht weiter untersucht.

Zusammenfassend kann festgestellt werden, dass geeignete Parametrierungen der unverzögerten und der verzögerten direkten Integration die besten Ergebnisse bringen, wobei hier die Kombination mit dem Complex-Verfahren besser abschneidet.

6.3 Aktorplatte

Bei der Aktorplatte verbieten sich auf Grund der langen Simulationszeiten vergleichende Läufe mit unterschiedlichen Parametrierungen. Stattdessen sind die bisher gemachten Erfahrungen (siehe auch [3]) praktisch anzuwenden. Verglichen werden Ergebnisse, die mit einer Parallelversion von GADO erzielt wurden [15], mit HyGADO unter Verwendung des Rosenbrock-Verfahrens bei direkter Integration und mittlerer Abbruchschranke, wobei die Hälfte der Startpopulation vorgeneriert und voroptimiert wurde [14]. Tabelle 3 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 viel sehr ähnliche Vektoren berechnet werden, die vom Filter als identisch bewertet werden. Dementsprechend wird in der Tabelle zwischen angeforderten und ausgeführten Simulationen unterschieden. HyGADO benötigt also nur etwa die Hälfte an real durchgeführten Simulationen, um ein besseres Ergebnis zu erzielen als die nichthybride Variante. GADO benötigte dazu 2 Tage und 15 Stunden auf 8 Suns unterschiedlicher Leistung, während HyGADO insgesamt 12 Tage und 16 Stunden auf einer Sun Ultra Sparc 10 lief. Bild 11 zeigt die 3 Designs mit ihrer recht unter-

schiedlichen Geometrie.

Tab.3: Optimierungsergebnisse

	Auslenkung		Spannung		Evaluationen	
	[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

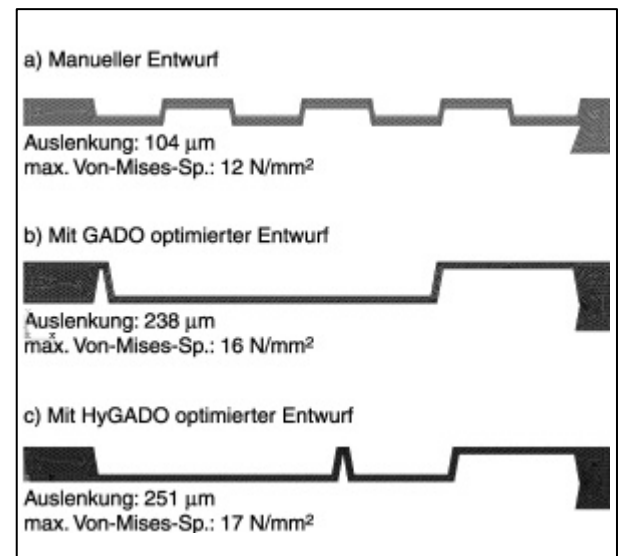


Abb.11: Aktorplatten: manueller Entwurf (a), Ergebnisse von GADO (b) und HyGADO (c)

7 Zusammenfassung und Ausblick

Die vorgestellten Anwendungen zeigen, dass die Kombination der allgemein anwendbaren lokalen Suchverfahren Rosenbrock und Complex mit dem Evolutionären Algorithmus GLEAM zu einem schnellen und robusten Optimierungswerkzeug führt. Da bei praktischen Anwendungen keine ausführlichen Untersuchungen wie bei den beiden relativ schnell simulierbaren Modellen des Drucksensors und des Heterodynempfängers möglich sind, muss eine Verfahrenskombination mit geeigneter Parametrierung gewählt werden. Zusammen mit den in [3] untersuchten Testfunktionen erscheint folgende Hybridisierung als geeignet: Verzögerte direkte Integration des Rosenbrock-Verfahrens, wobei die Wahl des Steuerungsparameter-Satzes keine allzu großen Auswirkungen hat. Schwieriger

ist dagegen die Bestimmung der Abbruchschranke des Rosenbrocks, da diese sehr anwendungsspezifisch ist. Eine mittlere Schranke bei gleichzeitiger Begrenzung der Iterationen auf wenige 1000 je nach Simulationsdauer erscheint hier als sinnvoll.

Als nächster Schritt ist die Parallelisierung von HyGADO in Vorbereitung, um so die noch recht langen Optimierungszeiten bei Verwendung von FEM-Modellen auf ein erträgliches Maß reduzieren zu können.

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: *HyGLEAM: Hybrid General purpose Evolutionary Algorithm and Method*. Proc. of 5th World Multiconference on Systemics, Cybernetics and Informatics (SCI'2001), Vol. III, pp.187-192, 2001.
- [4] C. Blume: *GLEAM - A System for Simulated Intuitive Learning*. Proc. of PPSN I, LNCS 496, Springer-Verlag, 1991.
- [5] W. Jakob, M. Gorges-Schleuter, I. Sieber: *Comparison of Evolutionary Algorithms for Design Optimization*. Proc. of PPSN V, LNCS 1498, Springer-Verlag, 1998.
- [6] 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
- [7] M. Gorges-Schleuter: *Genetic Algorithms and Population Structures - A Massively Parallel Algorithm*. Dissertation, Universität Dortmund, 1990.
- [8] M. Gorges-Schleuter: *A Comparative Study of Global and Local Selection in Evolution Strategies*. In: A. Eiben et al. (eds.): Proc. PPSN V, LNCS 1498, pp.367-377, Berlin: Springer, 1998.
- [9] M. J. Box: *A New Method of Constrained Optimization and a Comparison with other Methods*, Comp. J. 8, pp.42-52, 1965.
- [10] H. P. Schwefel: *Evolution and Optimum Seeking*. John Wiley & Sons; 1995.
- [11] D. Peters: *Unterstützung des Entwurfsprozesses von Mikrosystemen mit direkten Optimierungsverfahren*. Dissertation; Universität Bremen, 2001
- [12] H. H. Rosenbrock: *An Automatic Method for Finding the Greatest or Least Value of a Function*, Comp. J. 3, pp.175-184, 1960.
- [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, pp.65-70, 1998.
- [14] A. Quinte: *Modellierung und Optimierung der Aktorplatte*. Statusseminar des BMBF-Verbundprojekts OMID. Bremen, 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.