

MOMBES

Multiobjective Modelbased Evolution Strategy

Dany Meyer, Kratzer AG München
Carl-von-Linde-Str. 38, 85716 Unterschleißheim
Dany.Meyer@Kratzer-Automation.de

Forschungsbericht Künstliche Intelligenz FKI-246-02

Abstract

In der vorliegenden Arbeit wird ein System zur modellbasierten Mehrzieloptimierung unter Berücksichtigung von mehrdimensionalen Randbedingungen und Zielfunktionen vorgestellt. Es stützt sich insbesondere auf die Modellierung von Zielfunktionen und -Konflikten mit Hilfe Neuronaler Netze (NN) sowie einem Optimierungskern in Form einer speziell entwickelten Evolutionsstrategie (ES).

Im Gegensatz zu den bisher entwickelten multikriteriellen Evolutionären Algorithmen (MOEA) gestattet der in dieser Arbeit entwickelte Ansatz zum einen eine Extraktion von Wissen aus der Pareto-Front und nutzt zum anderen die Eigenschaft lernfähiger Neuronaler Netze zur Beeinflussung der Suchrichtung des Optimierungsalgorithmus. Die zusätzliche Kombination mit einem gradientenbasierten Optimierungsalgorithmus machen den Ansatz zu einem multi-hybrid-System, das sich durch sehr gute Konvergenzeigenschaften und eine hohe Qualität der generierten Paretofront auszeichnet. Als Besonderheit dieses Ansatzes sei die Hybridisierung von ES und NN genannt. Dabei wurde die Modellierung der Verteilung der Elemente der Pareto-Front in Form der Belehrung einer Selbstorganisierenden Merkmalskarte mit den Pseudo-Gewichten während und die Approximation der Pareto-Front durch RBF-Netze nach Abschluß der Optimierung als Unterstützung für das Decision-Making neu entwickelt.

1 Einleitung

Heute sind Aufgaben der Optimierung aus keinem Bereich der modernen Technik mehr wegzudenken. Dabei zeigt sich immer häufiger, daß es sich um komplexe Optimierungsprobleme handelt, die zum einen sich widersprechende Ziele und zum anderen eine große Anzahl von unterschiedlichen Randbedingungen enthalten. Diese Anforderungen stellen die Entwickler von modernen Optimierungsmethoden vor große Herausforderungen, die in den meisten Fällen nicht mehr allein mit Methoden der klassischen Mathematik lösbar sind. Vielmehr erfordern sie den zusätzlichen Einsatz lernfähiger Methoden, die durch das Ausnutzen von Synergien zwischen den klassischen und naturnahen Verfahren zur Entwicklung leistungsfähiger hybrider Systeme führen.

1.1 Multikriterielle Optimierungsprobleme

Im folgenden werden einige grundlegende Definitionen bezüglich der Lösung multikriterieller Optimierungsprobleme getroffen.

Definition 1: MOP:

Ein multikriterielles Optimierungsproblem (MOP: multi-objective problem) $F(\vec{x})$ wird definiert durch

- den Vektor $\vec{x} = (x_1, x_2, \dots, x_n) \in X$ von n Entscheidungsvariablen des Entscheidungsraumes (*decision space*) X ,
- den Vektor $\vec{f} = (f_1, f_2, \dots, f_k) \in F$ von k Zielfunktionen im Raum der Zielfunktionen (*objective space*) F , wobei die $f_i = f_i(\vec{x})$ Funktionen der Entscheidungsvariablen \vec{x} sind, sowie
- den Vektoren $\vec{g} = (g_1, g_2, \dots, g_m) \in G$ und $\vec{h} = (h_1, h_2, \dots, h_o) \in H$ von m bzw. o Randbedingungen (*constraints*) im Raum der Randbedingungsfunktionen (*constraint space*) G und H , mit $g_i = g_i(\vec{x})$ und $h_i = h_i(\vec{x})$ als Funktionen von \vec{x} .

Ohne Beschränkung der Allgemeinheit sei das generelle Optimierungsziel eines MOP's die *gleichzeitige* Minimierung¹ der Komponenten $f_i = f_i(\vec{x})$ des Vektors der Zielfunktionen \vec{f} unter Berücksichtigung der Randbedingungen $\vec{g}(\vec{x}) > \vec{0}$ und $\vec{h}(\vec{x}) = \vec{0}$:

$$\min \vec{f}(\vec{x}) = [f_1(\vec{x}), \dots, f_k(\vec{x})]^T \text{ mit } \vec{g}(\vec{x}) = [g_1(\vec{x}), \dots, g_m(\vec{x})]^T > \vec{0}, \vec{h}(\vec{x}) = [h_1(\vec{x}), \dots, h_o(\vec{x})]^T = \vec{0}.$$

In dieser Arbeit werden statische Optimierungsprobleme betrachtet, d.h. Optimierungsprobleme, bei denen die Lage des Optimums zeitinvariant ist. Methoden zur Lösung statischer Optimierungsprobleme sind jedoch auch für dynamische Optimierungsprobleme relevant, da deren Lösungsstrategie i.d.R. darin besteht, den betrachteten Zeitbereich in Abschnitte aufzuteilen und in jedem der so entstandenen Zeitfenster wiederum eine statische Optimierung für die sich veränderten Bedingungen durchzuführen.

Definition 2: feasible set

Die Menge der zulässigen Vektoren eines MOP $X_{feasible}$ oder X_L im nicht leeren Lösungsraum L ist definiert als die Menge der Vektoren der Entscheidungsvariablen \vec{x} , die den Randbedingungen \vec{g} und \vec{h} genügen: $X_{feasible} = \{\vec{x} \in X \mid \vec{g} > \vec{0} \wedge \vec{h} = \vec{0}\}$. Jedes \vec{f} aus F bestimmt eine Abbildung von $X_{feasible}$ in den Raum der Zielfunktionen

$$F_{feasible} : F_{feasible} = f(X_{feasible}) = \bigcup_{\vec{x} \in X_{feasible}} \{\vec{f}(\vec{x})\}.$$

¹ Dies ist möglich, da jedes Maximierungs- bzw. kombiniertes Maximierungs- bzw. Minimierungsproblem $F(\vec{x})$ durch einfache Konvertierung: $\min\{F(\vec{x})\} = -\max\{-F(\vec{x})\}$ in ein Minimierungsproblem umgeformt werden kann.

Entsprechend der Minima der einzelnen Komponenten $f_i = f(\vec{x})$ des Vektors der Zielfunktionen gelte für den Raum der Entscheidungsvariablen: $X_{feasible, i} = \left\{ \vec{x} \in X_{feasible} \mid f_i(\vec{x}) \min \right\}$ und $\vec{x}^{*i} := \min(f_i(\vec{x}))$; $x \in X_{feasible, i}$

Definition 3: Ideallösung

Haben alle Komponenten $f_i = f(\vec{x})$ ihr Minimum für ein und dasselbe \vec{x}^* , so spricht man von einer *idealen* Lösung \vec{x}_{ideal} .

Definition 4: Idealvektor

Der Idealvektor \vec{f}_{ideal} eines MOP's ist der Vektor, der als Komponenten i die Optimalwerte der Zielfunktionen $f_i(\vec{x})$ enthält: $\vec{f}_{ideal} = [f_1(\vec{x}^{*1}), f_2(\vec{x}^{*2}), \dots, f_k(\vec{x}^{*k})]$.

Definition 5: Zielfunktionskonflikt

Ein Zielfunktionskonflikt liegt vor, wenn zwei Optimierungsziele $f_i = f(\vec{x})$ und $f_j = f(\vec{x})$ disjunkt sind d.h. $X_{feasible, i} \cup X_{feasible, j} = \emptyset$ und $\vec{x}^{*i} \neq \vec{x}^{*j}$. Dann ist die Verbesserung eines Zielkriteriums i nicht ohne die Verschlechterung des Zielkriteriums j möglich und umgekehrt d.h. es ist nicht möglich, eine Lösung \vec{x} zu finden, die die Zielfunktionen f_i und f_j gleichzeitig minimiert.

Beispiel [1] illustriert ein MOP mit Zielfunktionskonflikt. Für dieses Beispiel existiert keine Ideallösung, der daher nicht erreichbare Idealvektor des Problems ist: $\vec{f}_{ideal} = [f_1(\vec{x}^{*1}), f_2(\vec{x}^{*2})] = [0,0]$ mit $\vec{x}^{*1} = [1,-1]$ und $\vec{x}^{*2} = [-1,1]$.

$$\min F(\vec{x}) = \vec{f}(\vec{x}) = [f_1(\vec{x}), f_2(\vec{x})] \tag{1}$$

$$f_1(\vec{x}) = 1 - e^{-(x_1-1)^2 - (x_2+1)^2}$$

$$f_2(\vec{x}) = 1 - e^{-(x_1+1)^2 - (x_2-1)^2}$$

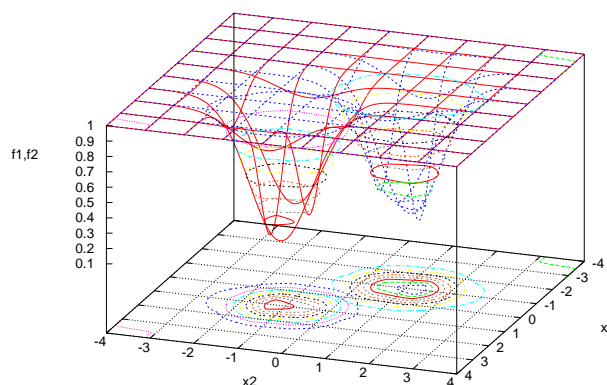


Abbildung 1.1: Visualisierung für das MOP [1]

Definition 6: best-over-all-Lösung

Die best-over-all-Lösung eines MOP's auf einer Lösungsmenge X' , ist diejenige Lösung $\vec{x}^B \in X'$, für die der euklidische Abstand zwischen $\vec{f}(\vec{x}^B)$ und dem Idealvektor \vec{f}_{ideal} minimal ist: $\vec{x}^B = \vec{x} \in X' \mid \min \|\vec{f}(\vec{x}) - \vec{f}_{ideal}\|$.

Liegt ein MOP mit Zielfunktionskonflikt vor, müssen unter Einbeziehung von Vorwissen und Präferenzen Kompromisse bzw. sogenannte „Trade-offs“ zur Zielerreichung der einzelnen Zielfunktionen gefunden werden. Die Menge der „Trade-offs“ oder der Lösungen, die in keiner Dimension des Vektors der Zielfunktionen dem jeweiligen Gütekriterium schlechter aber ihm in mindestens einer Dimension besser entsprechen als alle übrigen Lösungen, wird nach dem italienischen Gelehrten Vilfredo Pareto als pareto-optimal bezeichnet. Er untersuchte schon zu Ende des 19ten Jahrhunderts Optimierungsaufgaben mit sich widersprechenden Zielen im Bereich der Wirtschaftswissenschaften. Im folgenden sollen einige Begriffe aus dem Bereich der Pareto-Optimalität formal eingeführt werden.

1.2 Die Pareto-Optimalität

Als *pareto-optimale Lösungen* bezeichnet man alle Lösungen, die effizient sind d.h. von keiner anderen Lösung dominiert werden. Dominant ist eine Lösung u gegenüber einer Lösung v dann, wenn die Lösung u hinsichtlich des Zielerreichungsgrades in keiner Dimension schlechter und in mindestens einer Dimension besser ist als v .

Definition 7: Pareto-Dominanz

Ein Vektor $\vec{u} = (u_1, u_2, \dots, u_n)$ wird genau dann als dominant gegenüber dem Vektor $\vec{v} = (v_1, v_2, \dots, v_n)$ mit der Notation: $\vec{u} \prec \vec{v}$ bezeichnet, wenn \vec{u} partiell echt kleiner als \vec{v} ist, d.h. $\forall i \in \{1, \dots, n\}: u_i \leq v_i \wedge \exists i \in \{1, \dots, n\}: u_i < v_i$.

Dagegen dominiert \vec{u} den Vektor \vec{v} genau dann schwach (weakly dominates) und wird mit der Notation: $\vec{u} \preceq \vec{v}$ bezeichnet, wenn gilt: $\forall i \in \{1, \dots, n\}: u_i \leq v_i$

Die Vektoren \vec{u} und \vec{v} sind bezüglich der Pareto-Dominanz genau dann nicht vergleichbar (indifferent) und werden mit der Notation $\vec{u} \sim \vec{v}$ bezeichnet, wenn gilt: $\forall i \in \{1, \dots, n\}: \exists i \in \{1, \dots, n\}: u_i > v_i \wedge \exists i \in \{1, \dots, n\}: u_i < v_i$

Definition 8: Pareto-Optimalität

Eine Lösung \vec{x} sei genau dann pareto-optimal, wenn kein \vec{x}' existiert, für das der Vektor $\vec{v} = F(\vec{x}') = (f_1(\vec{x}'), f_2(\vec{x}'), \dots, f_n(\vec{x}'))$ dominant gegenüber dem Vektor $\vec{u} = F(\vec{x}) = (f_1(\vec{x}), f_2(\vec{x}), \dots, f_n(\vec{x}))$ ist.

Definition 9: Menge pareto-optimaler Lösungen (Pareto-Set)

Für ein definiertes MOP $F(\vec{x})$ sei die Menge aller pareto-optimaler Lösungen P^* definiert als: $P^* := \{\vec{x} \in X \mid \neg \exists \vec{x}' \in X : F(\vec{x}') \prec F(\vec{x})\}$

Während sich der Begriff der pareto-optimalen Lösungen auf den Raum der Entscheidungsvariablen bezieht, wird deren Abbildung in den Raum der Zielfunktionen als Pareto-Front bezeichnet.

Definition 10: Pareto-Front

Für ein definiertes MOP $F(\vec{x})$ und seine pareto-optimale Lösungsmenge P^* sei die Paretofront PF^* definiert als: $PF^* := \{\vec{u} = F(\vec{x}) = [f_1(\vec{x}), \dots, f_n(\vec{x})] \mid \vec{x} \in P^*\}$

Nach Horn [HornNafpl_1993] ist die Pareto-Front eines MOP's mit k Zielfunktionen, eine beschränkte Hyperfläche der Dimension $(k-1)$. Veldhuizen und Lamont beweisen in [VeldLa_1999], daß die Pareto-Front eines MOP's für $k=2$ Zielfunktionen höchstens eine beschränkte Kurve und für $k \geq 3$ Zielfunktionen höchstens eine beschränkte Hyperfläche ist.

Ist der Vektor der Entscheidungsvariablen \vec{x} eindimensional, d.h. die Zielfunktionen nur von einer Größe abhängig, ist die Pareto-Front analytisch berechenbar.

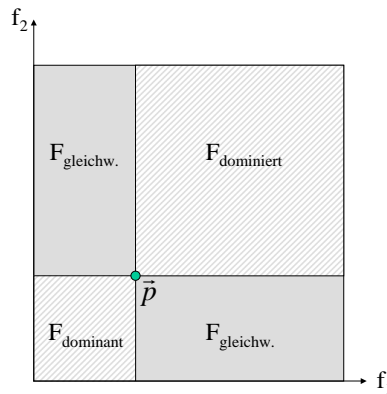


Abbildung 1.2: Pareto-Dominanz bezüglich des Punktes \vec{p} für ein MOP (simultane Minimierung von f_1 und f_2)

Entsprechend der obigen Definitionen gelte für Abbildung 1.2:

- Alle Vektoren $\vec{f} \in F_{\text{dominiert}}$ werden von \vec{p} dominiert d.h. \vec{p} ist dominant gegenüber jedem \vec{f} der Menge $F_{\text{dominiert}}$ d.h.: $\vec{f} \succ \vec{p} \mid \forall \vec{f} \in F_{\text{dominiert}}$.
- Alle Vektoren $\vec{f} \in F_{\text{dominant}}$ dominieren \vec{p} d.h. jedes \vec{f} der Menge F_{dominant} ist dominant gegenüber \vec{p} d.h.: $\vec{f} \prec \vec{p} \mid \forall \vec{f} \in F_{\text{dominant}}$.
- Alle Vektoren $\vec{f} \in F_{\text{gleichw.}}$ sind hinsichtlich der Pareto-Dominanz nicht vergleichbar mit \vec{p} .
- Die Lösung \vec{x}_p die \vec{p} bestimmt, ist hinsichtlich aller Lösungen $\vec{x}_{\text{dominiert}}$ die $\vec{f} \in F_{\text{dominiert}}$ bestimmen, pareto-optimal.
- Alle Lösungen $\vec{x}_{\text{dominant}}$ die $\vec{f} \in F_{\text{dominant}}$ bestimmen, sind bezüglich der Lösung \vec{x}_p die \vec{p} bestimmt, pareto-optimal.

2 Bekannte MOEA-Ansätze

Wie oben dargestellt ist es für die Lösung eines MOP's nicht ausreichend, eine einzelne Lösung zu generieren. Vielmehr ist es notwendig, eine Menge pareto-optimaler Lösungen zu erzeugen, unter denen der Anwender dann aufgrund von Präferenzen *seine* Optimallösung auswählt. Evolutionäre Algorithmen (EA), die aufgrund ihres Wirkprinzips auf einer Menge von Lösungen agieren, eignen sich für die Lösung eines solchen Problems besonders. Aber nicht nur die Anwendung des Populationsprinzips d.h. der massiv parallelen Suche machen EA für die Lösung von MOP's interessant, sondern auch ihre Fähigkeit, sich auch in stark zerklüfteten Zielfunktionslandschaften nicht in lokalen Minima zu fangen. Diese Eigenschaften zusammen mit der breiten Anwendbarkeit EA und ihres einfachen aber leistungsfähigen Grundprinzips machen sie zu hervorragenden Kandidaten zu Lösung multikriterieller Optimierungsprobleme. Auf die Darstellung Evolutionärer Algorithmen soll an dieser Stelle verzichtet, sondern statt dessen auf die Literatur verwiesen werden (z.B. [Bäck_1997] und [Schwefel_1995]).

Evolutionäre Algorithmen, die ein MOP lösen, werden nachfolgend als Multikriterielle Evolutionäre Algorithmen (MOEA: multi-objective evolutionary algorithm) bezeichnet.

A priori realisiert ein Standard-EA jedoch die Maximierung der Fitneß seiner Individuen, was einer Optimierung mit nur einer Zielfunktion entspricht. Es ist aus diesem Grunde also notwendig, ihn so anzupassen, daß er für die Lösung eines MOP's geeignet ist. Diese speziellen Techniken betreffen:

- *die Berechnung der Fitneß*: Hierarchische Optimierungsmethoden [Fourman_1985], Aggregationsansätze [Broek_1995] und [HajelaLin_1992], Pareto-basiertes Ranking [Goldberg_1989], [FonsFlem_1993],
- *die Berücksichtigung der „Ähnlichkeit“ von Lösungen*: Nischentechniken z.B. Fitneß Sharing [GoldRich_1987], Crowding [De_Jong_1975] sowie Mating Restriction [Goldberg_1989], [HajelaLin_1992], [FonsFlem_1993],
- *die Selektionsmechanismen von Individuen*: z.B. Selektion mit wechselnden Zielen [Schaffer_1984], [Kursawe_1991]
- *und das Populationskonzept selbst*: z.B. Multipopulationskonzept [Herdy_1992]

Eine sehr ausführliche und für den tieferen Einblick in das Thema MOEA empfehlenswerte Übersicht gibt K. Deb in [Deb_2002].

Im folgenden sollen die bekanntesten Vertreter von MOEA überblicksartig beschrieben werden, die für die vergleichenden Simulationen in dieser Arbeit verwendet wurden. Es wird in Abhängigkeit davon, ob zur Fitneßberechnung und Selektion das Konzept der Pareto-Dominanz verwendet wird, zwischen nicht pareto-basierten und pareto-basierten Ansätzen unterschieden. Zusätzlich werden zwei Vertreter hybrider MOEA beschrieben, deren Konzepte in der vorliegenden Arbeit weiterentwickelt wurden.

2.1 Nicht pareto-basierte Ansätze

2.1.1 „Vector Evaluated Genetic Algorithm“ (VEGA) von Schaffer

Der erste für ein multimodales Optimierungsproblem angepaßte EA wurde schon 1984 von J.D.Schaffer als Vector Evaluated Genetic Algorithm“ (VEGA) in [Schaffer_1984] und [Schaffer_1985] vorgestellt. Sein Ansatz setzt auf einem Standard-GA des GENESIS-System von Grefenstetten [Grefenst_1984] auf und erweiterte dessen Selektionsmechanismen in eine Selektion mit wechselnden Zielen. Die stochastische Selektion wurde bei k unterschiedlichen Zielfunktionen in k Teilschritte zerlegt: bei jedem der k Teilschritte wurde ein gleich großer

Anteil (k/N bei einer Populationsgröße N) von Individuen unter Berücksichtigung nur eines der k Zielkriterien fitnessproportional ausgewählt und in den mating-pool gespeichert. Alle anderen Mechanismen, wie Crossover und Mutation, werden standardmäßig ausgeführt und setzen auf diesem mating-pool auf.

Auf den ersten Blick erscheint diese Vorgehensweise erfolgversprechend, da Individuen, die mehreren Zielkriterien gut entsprechen, eine höhere Reproduktionswahrscheinlichkeit erhalten als Individuen, die nur für eine Zielfunktion minimale Werte generieren. Andererseits haben aber Individuen mittlerer Performance, d.h. die in keiner Zielfunktionsdimension minimal, aber in allen relativ nahe dem Minimum sind, weniger Chancen sich zu reproduzieren als überspezialisierte Individuen, die nur eine Zielfunktionsdimension minimieren. Mit diesem Ansatz können also die Eckpunkte der Pareto-Front eines MOP's relativ sicher gefunden werden, jedoch nicht zwangsläufig die dazwischen liegenden Kompromiß-Lösungen. In späteren Erweiterungen des VEGA-Systems wurde dieser Nachteil teilweise wieder ausgeglichen, indem spezielle Heuristiken zur „Durchmischung“ des Genpools angewandt wurden. Trotzdem stellt VEGA keinen echten pareto-basierten Ansatz dar, da er im mating-pool defacto eine Mittelwertbildung der Fitness entsprechend der einzelnen Teilziele realisiert:

... shuffling sub-populations together, or having different objectives affecting different tournaments, corresponds to averaging the fitness components associated with each of the objectives. [FonsFlem_1995] S. 8

Trotz dieser Nachteile hat das VEGA-System als erster Schritt bei der Entwicklung von MOEA eine große historische Bedeutung und erzielt bei Performance-Vergleichen gute Ergebnisse.

2.1.2 „Weighting-based Genetic Algorithm“ (HLGA) von Hajela und Lin

Ebenfalls einen nicht-pareto basierten Ansatz stellt der von P. Hajela und C.-Y. Lin in [HajelaLin_1992] beschriebene „Weighting-based Genetic Algorithm“ (HLGA) dar. Dort wird ein „weighted-sum“-Ansatz zur Aggregation der einzelnen normierten Zielfunktionen $f_i(\vec{x})$ verwendet sowie deren Gewichtungsfaktoren w_i kodiert als Entscheidungsvariablen mit in den Genotyp des Individuums aufgenommen. Um die Diversität der Kombinationen der Gewichtungsfaktoren zu gewährleisten, wird ein phänotypisches *sharing* auf die w_i eingesetzt. Auf diese Weise wird eine parallele Suche mehrerer Lösungen (entsprechend der unterschiedlichen Kombination der w_i) realisiert. Zusätzlich zum Sharing der Gewichtungsfaktoren wird *mating restriction* als weitere Nischentechnik verwendet, um auch die Diversität der Lösungen selbst zu erhöhen. Als Selektionsmethode wird Wettkampfselektion eingesetzt.

Vorteilhaft bei dieser Methode ist, daß bei der Auswahl der Lösung die entsprechend berechneten w_i mit angegeben werden können – allerdings ist der praktische Nutzen dieser Informationen nicht sehr hoch anzusetzen.

Innerhalb weniger Generationen werden mit diesem Ansatz gute Ergebnisse erzielt. Allerdings konvergiert er selbst bei großen Populationen sehr rasch auf eine einzelne Lösung und ist sehr sensitiv gegenüber Parametereinstellungen (z.B. dem Nischenradius). Da die Zielfunktionswerte normiert aggregiert werden, muß der Anwender eine ungefähre Vorstellung über deren Wertebereich besitzen.

2.2 Pareto-basierte Ansätze

2.2.1 „Niched Pareto Genetic Algorithm“ (NPGA) nach Horn und Nafpliotis

J.Horn und N.Nafpliotis kombinieren in ihrem in [HornNafpl_1993] vorgestellten „Niched Pareto Genetic Algorithm“ (NPGA) das Konzept der Pareto-Dominanz mit einer erweiterten Form der Wettkampfselektion. Bei der von ihnen verwendeten Selektion werden stochastisch sowohl zwei Wettkampfkandidaten, als auch eine Vergleichsmenge von Individuen aus der Population ausgewählt. Die Größe der zum Vergleich benutzten Subpopulation bestimmt den Selektionsdruck und wird durch den Parameter t_{dom} (i.d.R. in %-der Populationsgröße N angegeben) definiert. Wird einer der beiden Wettkampfkandidaten von keinem Individuum der Vergleichspopulation dominiert, geht er als Sieger der Selektion hervor. Werden dagegen beide Kandidaten dominiert oder dominieren alle Individuen der Vergleichspopulation, wird der Kandidat ausgewählt, in dessen phänotypischer Nische (definiert durch σ_{share}) sich die wenigsten Individuen der Vergleichspopulation befinden.

Der NPGA gilt bei nicht zu groß gewählten σ_{share} als einer der schnellsten MOEA, der in der Lage ist, eine den Raum $X_{feasible}$ weit abdeckende Pareto-Front zu generieren, jedoch in seiner Standardform keine Randbedingungen berücksichtigen kann. Nachteilig an diesem Ansatz ist, daß sich die beiden Parameter t_{dom} und σ_{share} sehr sensitiv auf die Konvergenz und Gestalt der Pareto-Front auswirken.

2.2.2 „Nondominated Sorting Genetic Algorithm“ (NSGA und NSGA2) nach Srinivas und Deb

N. Srinivas und K. Deb beschreiben in [SrinDeb_1994] einen weiteren pareto-basierten MOEA, den „Nondominated Sorting Genetic Algorithm“ (NSGA). In ihm wird das von Goldberg in [Goldberg_1989] vorgeschlagene Schema zur Berechnung des Pareto-Rangs erweitert. Auch hier erfolgt die Berechnung der Fitness der Individuen in mehreren Stufen. Zunächst werden nach dem Schema von Goldberg alle pareto-dominanten Individuen identifiziert, zu einer Front zusammengefaßt und erhalten einen temporären Fitnesswert, der proportional zur Populationsgröße ist. Anschließend erfolgt zwischen den Individuen dieser ersten Front ein phänotypisches Sharing im Raum der Entscheidungsvariablen unter Berücksichtigung eines definierten σ_{share} - es wird also Sharing nur zwischen Individuen gleichen Rangs durchgeführt. Anschließend wird die temporäre Fitness auf einen Wert gesetzt, der kleiner als der kleinste durch das Sharing entstandene Fitnesswert ist und der Vorgang wiederholt sich für die folgende Front, ohne Berücksichtigung der Individuen, denen bereits ein Fitnesswert zugeordnet wurde. Zusätzlich verwenden Deb und Srinivas eine stochastische *remainder*-proportionale Selektion.

Kritiker dieses Ansatzes kennzeichnen die Reduktion der einzelnen Zielfunktionswerte auf die temporäre Fitness als nachteilig [Coello_1996]. Der NSGA ist in der Lage, eine PF^* zu generieren, diese sollte jedoch extern gespeichert werden, da sie nicht stabil über eine große Anzahl von Generationen (mehr als 100) innerhalb der Population repräsentiert wird. Durch das phänotypische Sharing im Raum der Entscheidungsvariablen, ist es möglich Lösungen zu generieren, die für identische Zielfunktionen eine unterschiedliche Kombination der Entscheidungsvariablen zulassen, was für den Praktiker von großer Bedeutung sein kann. Auch bei diesem Ansatz gilt als Kritikpunkt, daß die Performance des Algorithmus' von der Wahl des Nischenradius σ_{share} abhängt, der problemabhängig justiert werden muß.

Eine aktuelle Erweiterung des NSGA ist der ebenfalls von Deb et. al. in [Deb_2000] vorgestellte NSGA2. In ihm wird im Gegensatz zum NSGA das Eliteprinzip verwendet und damit u.a. die

Ergebnisse von Zitzler (siehe Abschnitt 2.2.4; Eliteselektion begünstigt die Herausbildung von pareto-dominanten Individuen) berücksichtigt. So wird die jeweils neue Generation nicht mehr wie bisher ausschließlich aus der Menge der Nachkommen, sondern aus der Vereinigungsmenge der Eltern- und Nachkommenindividuen generiert.

Das aufgrund seiner Parameterabhängigkeit kritisierte Niching wird durch einen erweiterten Selektionsmechanismus ersetzt, bei dem die Nachbarschaft von Individuen mit identischem Pareto-Level (d.h. Individuen einer Pareto-Front, siehe oben) durch eine sogenannte *crowding distance* berücksichtigt wird. Die *crowding distance* eines Individuums \vec{a}_i berechnet sich aus der mittleren euklidischen Distanz von $\vec{f}(\vec{a}_i)$ zu seinen unmittelbaren Nachbarindividuen über alle Dimensionen von $\vec{f}(\vec{x})$ (bei $k = 2$ entspricht das der mittleren Seitenlänge eines Rechtecks. Um eine möglichst homogene Verteilung der Elemente der Pareto-Front zu erreichen, werden bei der Selektion diejenigen Individuen begünstigt, deren *crowding distance* größer ist.

Durch den Einsatz von schnellen Sortieralgorithmen, konnte die Komplexität des Algorithmus' gesenkt und damit seine Effizienz gesteigert werden.

Der NSGA2-Algorithmus zählt heute, neben dem SPEA2-Algorithmus zu den effektivsten MOEA.

2.2.3 „Multiobjective Genetic Algorithm“ (MOGA) nach Fonseca und Fleming

Auch Fonseca und Fleming verwenden in ihrem „Multiobjective Genetic Algorithm“ - MOGA [FonsFlem_1993] - zur Berechnung der Fitneß eines Individuums den pareto-basierten Rang und stützen sich ebenso wie Srinivas und Deb beim NSGA auf das Schema von Goldberg. Allerdings berechnet sich die Fitneß der Individuen hier nicht direkt aus ihrem Pareto-Rang, sondern hängt von der Anzahl der von ihm dominierten Individuen innerhalb der Population ab, was zu einer volleren und glatteren PF^* führt. Als Selektionsmechanismus wird Wettkampfselektion eingesetzt (i.d.R. mit jeweils 2 Kandidaten). Um zusätzlich die Diversität der Lösungen innerhalb der Population zu erhöhen, wird im Raum der Zielfunktionen *sharing* verwendet.

Fonseca und Fleming zeigen in ihrem Ansatz auch die Möglichkeit, Zielpräferenzen in Form eines Goal-Vectors zu integrieren und so den Focus auf definierte Bereiche der PF^* zu lenken [FonsFlem_1998].

Auch bei diesem Ansatz stellt sich die Frage nach der richtigen Wahl des Nischenradius, wenngleich seine Wahl keinen zu großen Einfluß auf die Performance des Algorithmus besitzt. Durch Sharing im Raum der Zielfunktionen ist es nicht möglich, zwei unterschiedliche Lösungen \vec{x} mit identischen Zielfunktionswerten zu halten, was aber meist genau das Ziel des Anwenders ist.

2.2.4 „Strength Pareto Evolutionary Algorithm“ (SPEA und SPEA2) von Zitzler

Der „Strength Pareto Evolutionary Algorithm“ (SPEA) von Zitzler ([ZitzlerTh_1998] und [Zitzler_1999]) geht im Gegensatz zu den bisher vorgestellten MOEA streng von einem Maximierungsproblem aus. Auch Zitzler verwendet als Basis der Fitneßberechnung den Pareto-Rang von Lösungen. Jedoch werden nach jeder Generation alle Individuen mit dem Rang=1 aus der Population P_t entfernt und mit ihnen eine externe Elitemenge, das sogenannten Pareto-Set \bar{P}_t , aktualisiert. Das Pareto-Set enthält nur dominante Lösungen. Damit die Größe dieser externen Pareto-Menge eine definierte Anzahl von Elementen nicht überschreitet, verwendet Zitzler ein Clusterverfahren. Ist das Pareto-Set aktualisiert, erhält jedes seiner

Elemente p_j^f in Abhängigkeit davon, wievielen Individuen der Population es zugeordnet werden kann (siehe unten) eine Maßzahl, die *strength* s_j (siehe Gleichung [2] mit n als Anzahl der Individuen \vec{a}_i , die p_j^f zugeordnet werden können und N als Populationsgröße von P_t).

$$s_j = \frac{n}{N+1} \quad [2]$$

Die Fitneß $\Phi(\vec{a}_i)$ des Individuums \vec{a}_i der Population berechnet sich nun in Abhängigkeit davon, von wievielen Elementen j des Pareto-Sets \bar{P}_t es zugeordnet wird (siehe Gleichung [3]).

$$\Phi(\vec{a}_i) = 1 + \sum_{j, j < i} s_j \quad [3]$$

Die „Zuordnungs“-Relation eines Elements des Pareto-Sets p_j^f läßt sich geometrisch so erklären, daß alle Individuen \vec{a}_i , deren entsprechende Tupel $p_i := [f_{i1}, f_{i2}, \dots, f_{ik}]$ von dem Hyperquader, der das Tupel $p_j^f := [f_{j1}, f_{j2}, \dots, f_{jk}]$ zu einem definierten Referenzpunkt im Raum der Zielfunktionen aufspannt, eingeschlossen werden, dem Element p_j^f zuzuordnen sind. Bei Maximierungsproblemen ist der oben genannte Referenzpunkt i.d.R. der Koordinatenursprung im Raum der Zielfunktionen.

Nach der Berechnung der Fitneß erfolgen Selektion und Reproduktion nach dem Standard-GA, wobei die Selektion (Wettkampfselektion) auf der Vereinigungsmenge der Population und dem externen Pareto-Set aufsetzt. Der große Vorteil von SPEA ist sicherlich, daß es gelungen ist, die Fitneß der Individuen unter Berücksichtigung von Nachbarschaftsbeziehungen zu berechnen, *ohne* einen vom Anwender zu justierenden Parameter (wie den Nischenradius beim Sharing) einzusetzen. Zusätzlich gehen durch die Anwendung des Eliteprinzips bei der Speicherung der dominanten Individuen im Pareto-Set, keine einmal berechneten pareto-optimalen Lösungen verloren und wird durch die Anwendung eines Clusterverfahrens eine gleichmäßig verteilte PF^* generiert.

Eine aktuelle Weiterentwicklung des SPEA-Algorithmus wird mit dem SPEA2-Algorithmus in [ZitzLauTh_2001] vorgestellt. Die Erweiterungen betreffen die Berechnung der Fitneß der Individuen und den Auswahlalgorithmus, der auf dem Pareto-Set \bar{P}_t angewandt wird.

Bei SPEA2 wird bei der Berechnung der Fitneß der Individuen aus P_t zusätzlich zur *strength* s_j nach Gleichung [3] auch die Dichte der Population, in Form eines Dichtemaß $D(i)$, additiv berücksichtigt. Dabei wird $D(i) \in [0,1]$ nach dem „nearest neighbor“-Verfahren berechnet. So wird Individuen, die sich in einer Region hoher Dichte der Population befinden und denen nach Gleichung [3] u.U. die selbe Fitneß zugeordnet wurde wie Individuen in Regionen geringerer Dichte, jetzt eine geringere Fitneß zugewiesen. Auf diese Weise ist eine noch feinere Berechnung der Fitneß möglich, was zu einer Erhöhung der Variabilität der Individuen innerhalb der Population führt.

In SPEA2 wird der Clusteralgorithmus, der in SPEA zur Reduzierung der Anzahl der Elemente im Pareto-Set angewandt wird, durch einen einfachen Selektionsalgorithmus ersetzt: ist die Anzahl der dominanten Individuen größer als die maximal zulässige Anzahl der Individuen im Pareto-Set, werden diejenigen Individuen mit dem geringsten Dichtemaß aus P_t entfernt.

2.2.5 „Pareto Envelope-based Selection Algorithm“ (PESA) von Corne und Knowles

Der „Pareto Envelope-based Selection Algorithm“ (PESA) von Corne und Knowles stellt eine Weiterentwicklung der „Pareto Archived Evolution Strategy“ (PAES) von Corne und Knowles [KnowCorn_2000] dar und verwendet zur Berechnung der Fitness der Individuen eine sogenannte hyper-grid-Strategie. Dabei wird der Lösungsraum durch ein Hyper-Gitter in gleichmäßige Segmente aufgeteilt. Jedes Individuum erhält – ähnlich wie die strength bei SPEA – eine zusätzliche Maßzahl: den sogenannten *squeeze*-Faktor. Diese Maßzahl entspricht der Anzahl der Lösungen, die sich innerhalb desjenigen Hyper-Gitter-Segments befinden, daß dem Individuum zugeordnet werden kann. Ähnlich wie SPEA, arbeitet auch der EA bei PAES und PESA auf zwei Populationen: einer internen Population *IP* und einer externen Population *EP*, die der Archivierung der dominanten Individuen dient. Die Größe der externen Population wird i.d.R. deutlich höher als die der internen Population gewählt z.B. im Verhältnis 10:1. Nach jedem Generationszyklus wird *EP* mit den dominanten Lösungen aus *IP* aktualisiert. Übersteigt die Anzahl der Individuen in *EP* einem bestimmten Wert, werden solange diejenigen Individuen mit dem größten *squeeze*-Faktor aus der externen Population entfernt, bis die erforderliche Größe der Population erreicht wurde. Die interne Population wird nach jeder Generation vollständig gelöscht und durch Nachkommen der folgenden Generation ersetzt. Die Selektion der Eltern im Generationszyklus erfolgt daher generell nur aus *EP*, d.h. aus der Menge der dominanten Lösungen. Es wird Wettkampfselektion verwendet, wobei dasjenige Individuum selektiert wird, das den kleineren *squeeze*-Faktor besitzt. Crossover und Mutation werden nach dem Standard-GA durchgeführt.

PESA verwendet, ähnlich wie SPEA, konsequent das Eliteprinzip. Damit können zwar einmal gefundene dominante Lösungen nicht mehr verloren gehen; jedoch besteht die Gefahr vorzeitiger Konvergenz durch die Verringerung der Variabilität der Population. Die Sensitivität des Algorithmus gegenüber der Parametrierung des Hyper-Gitters stellt einen weiteren Nachteil dieses Ansatzes dar. Trotzdem konnten bei Benchmarks [CorKnoOa_2000] und [ZitzLauTh_2001] gute Ergebnisse im Vergleich mit SPEA erzielt werden.

2.3 Ausgewählte Hybride MOEA

2.3.1 Kombination zwischen MOEA und lokalen Suchstrategien

Bei der Kombination der MOEA mit lokalen Suchalgorithmen wird zwischen zwei Strategien unterschieden:

1. so kann bereits *während* der evolutionären Suche nach bestimmten Kriterien ein lokaler Suchalgorithmus angestoßen oder
2. erst *nach* Abschluß der Optimierung auf Individuen der Endpopulation angewandt werden.

Beide Herangehensweisen haben ihre Vor- und Nachteile. Häufig wird bei Methoden der ersten Strategie der Vorteil eines besseren Konvergenzverhaltens gegen den Nachteil eines größeren Rechenaufwandes aufgehoben. Unter Umständen „fangen“ sich die lokalen Suchoperatoren in stark zerklüfteten Fitnesslandschaften schon zu früh in lokalen Minima – ein Verhalten, das ja gerade durch den Einsatz EA minimiert werden soll. Den Kriterien, wann und auf welche Individuen die lokalen Suchoperatoren angewandt werden oder wann es effizienter ist, den MOEA ohne lokale Suche anzuwenden, kommt daher eine entscheidende Bedeutung zu. Da EA jedoch nicht stetig konvergieren, ist die Entscheidung, ab wann eine vorzeitige Konvergenz vorliegt schwierig. Aus diesem Grunde erfolgt die Auswahl der Individuen, auf die die lokalen Suchstrategien angewandt werden i.d.R. pro Generation gleichverteilt zufällig nach einer

Wahrscheinlichkeit, die im einfachsten Fall für alle Individuen gleich oder in Abhängigkeit von der Fitneß der Individuen unterschiedlich hoch sein kann.

Die zweite Strategie – d.h. die Anwendung der lokalen Operatoren auf die Elemente der Pareto-Front – nach Abschluß der Optimierung anzuwenden umgeht zwar die obengenannten Nachteile, bringt den MOEA jedoch auch um die Möglichkeit quasi große Evolutionssprünge während der Optimierung durchzuführen. Die lokalen Operatoren haben hier den Effekt, ein Finetuning zu realisieren d.h. die Elemente der Pareto-Front, die sich quasi am Rande eines pareto-optimalen Tales befinden, in dieses hineinzuziehen. An dieser Stelle sollen zwei Vertreter für Hybride-MOEA mit lokalen Suchstrategien genannt werden:

So kombinieren beispielsweise D.Quagliarella und A.Vincini in [QuagliaVic_1997] den MOGA-Algorithmus mit einem lokalen „hill-climbing“-Operator², der während der evolutionären Suche mit einer gewissen Wahrscheinlichkeit auf Individuen der aktuellen Population aufsetzt. Da seine Auswahlwahrscheinlichkeit von der Fitneß der Individuen und damit ihrer Dominanz abhängt, werden pareto-optimale Individuen bevorzugt in die lokale Suche einbezogen. Individuen, die dominiert werden, werden jedoch von der Anwendung der lokalen Suche nicht vollständig ausgeschlossen, sondern erhalten lediglich eine geringere Auswahlwahrscheinlichkeit.

K.Deb und T.Goel verwenden hingegen für ihren hybriden Ansatz in [DebGoel_2001] den NSGA2-Algorithmus, der ebenso wie bei D.Quagliarella und A.Vincini mit einem lokalen „hill-climbing“-Operator verknüpft wird, der jedoch erst nach Durchlaufen des MOEA auf jedes Element der Paretofront angewandt wird. Dabei wurde die Paretofront zuvor ähnlich wie beim SPEA-Algorithmus durch ein Clusterverfahren konvertiert. Bei der anschließenden Anwendung des lokalen Suchoperators auf den Elementen der Paretofront sollten sich idealerweise die einzelnen Elemente der Paretofront entlang eines hochdimensionalen Korridors in Richtung tatsächlicher Paretofront bewegen und dabei ihren relativen Abstand zueinander möglichst einhalten. Die Gefahr bei diesem Ansatz besteht darin, daß es u.U. nach Anwendung des „hill-climbing“-Operators zu einer Verklumpung der einzelnen Elemente der Pareto-Front kommt und damit ihre Ausbreitung im Raum der Zielfunktionen verringert wird.

2.3.2 Kombination zwischen MOEA und Methoden der Computational Intelligence

An dieser Stelle sollen hybride MOEA erwähnt werden, die die Kombination eines MOEA mit Methoden der Computational Intelligence realisieren.

Hierzu gehören u.a. MOEA in Kombination mit Neuronalen Netzen („Neuro-EA“). Es existieren heute eine große Anzahl verschiedener Hybridisierungsmöglichkeiten zwischen Neuronalen Netzen und EA. Die überwiegende Mehrzahl der Ansätze verwendet hierbei EA um optimale Netzwerkstrukturen oder –Parameter zu finden oder während des Lernprozesses die optimalen Lernregeln auszuwählen bzw. die Kopplungsstärken der Gewichte selbst auf die Trainingsdaten optimal anzupassen. In vielen Arbeiten werden NN auch zur Berechnung der Gütefunktionen und Verletzung der Randbedingungen genutzt. Allen diesen Ansätzen ist gemeinsam, daß sie NN als zusätzliche, vom eigentlichen Optimierungsprozeß abgekoppelte Module verwenden.

Einen anderen Ansatz verfolgen diejenigen Arbeiten, die NN direkt in die evolutionäre Suche einbeziehen. Das kann geschehen, indem die Struktur des NN selbst für die evolutionäre Suche genutzt wird: z.B. indem die Entscheidungsvariablen der aktuellen Generation als input-Schicht über die hidden-Schicht, die der Generation der Nachkommen entspricht, auf die output-

² I.d.R. werden unter diesem Begriff klassische, meist gradientenbasierte Optimierungsverfahren verstanden. Um diesen Ansatz lokal auf ein Individuum anwenden zu können, wird das MOP durch geeignete Aggregation der Zielfunktionswerte in ein Optimierungsproblem mit nur einer Zielfunktion (SOP) überführt.

Schicht, deren Neurone den einzelnen Zielfunktionen zugeordnet werden, abgebildet werden. Eine Realisierung dieser Idee stellt das Nussy-System von M.Köppen und S.Rudlof [KoepRudl_1998] dar.

Eine weitere Möglichkeit besteht darin, das NN dafür zu nutzen den Evolutionspfad zu lernen und die Reproduktionsoperatoren entsprechend des gelernten Wissens anzupassen. Diesen Ansatz verfolgen beispielsweise D.Büche et al. [BüMiKou_2002] mit ihrem SOM-MOEA, der in diesem Abschnitt näher dargestellt wird. Sie gehen in ihrer Arbeit von einem NSGA2- bzw. SPEA2-Algorithmus aus und verknüpfen diesen mit einem Neuronales Netz - genauer einer selbstorganisierenden Merkmalskarte (SOM³). In ihrer Arbeit erweitern sie das von M.Milano et al. in [MilSchKou_2001] vorgestellte Konzept, eine SOM für Rekombinationsoperatoren eines EA zu verwenden, auf multikriterielle Optimierungsprobleme. So belehren Büche et al. eine SOM nach jeder Generation mit den Entscheidungsvariablen der Elterngeneration der aktuellen Population. Damit wird die Dimension der Trainingsvektoren für die SOM der Anzahl der Entscheidungsvariablen des MOP's gleichgesetzt. Die Dimension der Kopplungen innerhalb des Gitters (Kohonenkarte) entspricht der Dimension der Paretofront, die bei k Zielfunktionen die Dimension $(k - 1)$ besitzt. Die Anzahl der Neurone auf dem Gitter variiert in Abhängigkeit von der Dimensionalität der Zielfunktionen (Büche et al. geben für eine eindimensionale Kohonenkarte, also für 2 Zielfunktionen, 20 Neurone an). Die Belehrung der Kohonenkarte erfolgt nach den Standardverfahren für SOM [Kohonen_2001]. Sowohl die verwendeten Mutations- als auch Rekombinationsoperatoren des MOEA setzen während der Generationszyklen auf der sich ständig aktualisierenden SOM auf. So wird bei der Rekombination zwischen zwei benachbarte Gitterpunkte innerhalb der Kohonenkarte interpoliert. Das entspricht einer intermediären Rekombination auf der Elterngeneration. Die Mutation hingegen extrapoliert mit einer bestimmten Schrittlänge auf Vektoren von Entscheidungsvariablen die sich außerhalb der Kohonenkarte abbilden würden. Die Schrittlänge wiederum hängt sowohl von einem Lernfaktor als auch einer Zufallskomponente ab.

Büche et al. geben als Vorteil ihrer Methode an, daß es mit Hilfe der SOM gelingt, zwei „ähnliche“ Individuen für die Rekombination auszuwählen. Diese Herangehensweise erscheint jedoch nur zum Ende der Evolution hin sinnvoll. Gerade die Rekombination von unterschiedlichen Individuen führt i.d.R. zu großen „Evolutionssprüngen“, die so verhindert werden. Ein weiterer Kritikpunkt ist, daß für die Belehrung der SOM lediglich die Entscheidungsvariablen, nicht jedoch die Zielfunktionen verwendet werden. Eine Ähnlichkeitsbeziehung im Raum der Entscheidungsvariablen, die bei diesem Ansatz in einer Nachbarschaftsbeziehung auf der Kohonenkarte umgesetzt wird, impliziert hier auch eine Ähnlichkeit im Raum der Zielfunktionen, was nicht zwangsläufig der Fall ist. Benachbarte Individuen auf der Kohonenkarte entsprechen also nicht auch Individuen mit ähnlichen Zielfunktionswerten. Das Ziel der multikriteriellen Optimierung besteht darin, eine homogene Paretofront zu generieren die sich möglichst dicht an die tatsächliche Paretofront anschmiegt. Gerade der Aspekt der Homogenität der Paretofront kann mit diesem Ansatz nicht zwangsläufig unterstützt werden. Das Auffinden von „Löchern“ auf der Paretofront wird durch diesen Ansatz nicht explizit unterstützt. Die SOM übernimmt hier lediglich die Funktion des Sortierens der Individuen bezüglich der Entscheidungsvariablen d.h. dem Auffinden von Nachbarschaftsrelationen, was nicht ihrer Leistungsfähigkeit gerecht wird.

Der Einsatz NN, die mit der aktuellen Population belehrt werden und es dem EA gestatten, dieses Wissen für die evolutionäre Suche auszunutzen, wird aus Sicht des Autors als sehr sinnvoll eingeschätzt. In [Meyer_2000] wurde diese Idee bereits formuliert und mit dem in dieser Arbeit vorgestellten Ansatz „MOMBES“ konsequent weiterentwickelt. Statt wie bei dem SOM-

³ Zur Struktur und Lernalgorithmen von SOM siehe [Kohonen_2001]

MOEA von Bücher et al. „ähnliche“ Individuen zu rekombinieren, erscheint es effektiver, das durch die NN repräsentierte Wissen über die Population bzw. der aktuellen Paretofront dazu zu nutzen, um Regionen geringer Dichte auf der Paretofront systematisch zu verringern. Genau dieser Gedanke wurde mit dem MOMBES-System realisiert.

Natürlich gibt es noch eine große Menge weiterer Möglichkeiten der Verknüpfung MOEA mit anderen, naturnahen Optimierungsmethoden. Da im Rahmen dieser Arbeit bezüglich der Hybridisierung jedoch gradientenbasierte lokale Suchalgorithmen und Neuronale Netze als Modellbildungskomponenten im Vordergrund stehen, sollen weitere Möglichkeiten der Kombination MOEA mit naturnahen Verfahren der Computational Intelligence nur der Vollständigkeit halber erwähnt werden.

- MOEA in Kombination mit Methoden der fuzzy-Logik: [GomJimIlb_1998], [Gueugn_1997] und [IshMur_2001]
- MOEA in Kombination mit tabu search: [Hansen_1997], [GomJimIlb_1998] und [Know_2002]
- MOEA in Kombination mit simulated annealing: [Know_2002]

Die Verbindung zwischen MOEA und lernfähigen Systemen erscheint dem Autor deshalb als sehr sinnvoll weil es möglich ist:

- mittels der NN die Evolutionsrichtung zu lernen und damit die evolutionäre Suche zielgerichteter zu gestalten und damit zu beschleunigen sowie
- ein Einsatz NN es dem Anwender gestatten, Wissen aus dem Evolutionsprozeß zu extrahieren und damit den MOEA von einer „black box“- in eine „gray-box“-Methode zu überführen.

Beide Argumente wurden in den vorliegenden Arbeit in den realisierten modellbasierten MOEA (MOMBES) verarbeitet.

3 Multiobjective Modelbased Evolution Strategy

In diesem Kapitel wird das System MOMBES: **M**ulti **O**bjective **M**odel**B**ased **E**volution **S**trategy detailliert beschrieben. Dabei werden zunächst die Anforderungen an das Verfahren und seine Abgrenzung zu den in Abschnitt 2 beschriebenen MOEA formuliert.

3.1 Anforderungen an das Verfahren

Es sei ein robustes, schnelles und leicht auf neue, multikriterielle Optimierungsprobleme anpaßbares System zu erarbeiten, das die interaktive Rolle des Anwenders unterstreicht. Das System soll sowohl nichtlineare Zielfunktionen minimieren als auch lineare und nichtlineare Restriktionen verarbeiten können. Es sind Lösungen, die den Randbedingungen genügen denjenigen Lösungen vorzuziehen, die die Zielfunktionen minimieren. Die Anzahl der Randbedingungen und Zielfunktionen soll dabei nicht beschränkt sein. Es soll keine exakte Gewichtung der Zielfunktionen zu Beginn der Optimierung angegeben werden. Statt dessen ist eine Menge von maximal \bar{N} Kompromißlösungen (Pareto-Set) zu generieren, deren entsprechende Zielfunktionsstapel (Elemente der Pareto-Front) möglichst gleichmäßig im Raum der Zielfunktionen verteilt sind. Während der Iteration soll die evolutionäre Suche zielgerichtet in Richtung homogen verteilter Pareto-Front beeinflußt werden. Dazu werden Informationen aus denen zum aktuellen Iterationsschritt vorliegenden dominanten Lösungen verwendet. Nach Abschluß der Optimierung soll interaktiv eine Lösung aus dem Pareto-Set ausgewählt werden können. Durch Anwendung eines Interpolationsverfahrens soll zusätzlich die Möglichkeit bestehen, auch auf Lösungen zuzugreifen, deren Zielfunktionsstapel sich zwischen den Elementen der Pareto-Front befinden. Auf diese Weise kann die für den Anwender „ideale Lösung“ interaktiv nach Abschluß der Optimierung gefunden werden. Es soll weiterhin möglich sein, interaktiv qualitative Aussagen über den Einfluß einzelner Entscheidungsvariablen auf einzelne Zielfunktionen sowie über eventuelle Abhängigkeiten der Zielfunktionen untereinander zu extrahieren.

3.2 Allgemeines Ablaufschema

Um die oben beschriebenen Anforderungen erfüllen zu können, wurde ein modellbasierter, multi-hybrider MOEA realisiert, der sich insbesondere auf die Kombination von Evolutionsstrategien (ES) und Neuronalen Netzen (NN) stützt.

Abbildung 3.1 zeigt das allgemeine Ablaufschema zur Modellbasierten Mehrzieloptimierung und die Einordnung von MOMBES in die Problematik der datengetriebenen Optimierung in der Prozeßindustrie.

Gegeben sei ein Prozeß, der sich durch eine Menge von Tupeln aus Eingangs- und Ausgangsgrößen (Beispielsituationen) oder spezielles Wissen in Form von Regeln und Berechnungsvorschriften über den Zusammenhang zwischen Eingangs- und Ausgangsgrößen darstellen kann (A). In Abhängigkeit davon, in welcher Form das Wissen über die Zusammenhänge zwischen Eingangs- und Ausgangsgrößen vorliegt, kann die Berechnung der Zielfunktionen und Randbedingungen mit Hilfe verschiedenster Modelle (White-Box, Gray-Box oder Black-Modelle) erfolgen. Liegt das Prozeßmodell vor, kann der Anwender flexibler Modellversuche durchführen, als dies am realen Prozeß möglich ist (C). So können Erkenntnisse über das Prozeßverhalten in unbekanntem Betriebspunkten gewonnen und verschiedenste Prozeßsituationen simuliert werden, deren Durchführung in der Realität unter Umständen zu gefährlich oder zu komplex ist. Durch Erkenntnisse, die durch das Prozeßmodell gewonnen werden, können gezielt spezielle Arbeitspunkte des Modells validiert werden

(Versuchsplanung) und durch abermaliges Belernen des Modells seine sukzessive Verfeinerung erreicht werden (D).

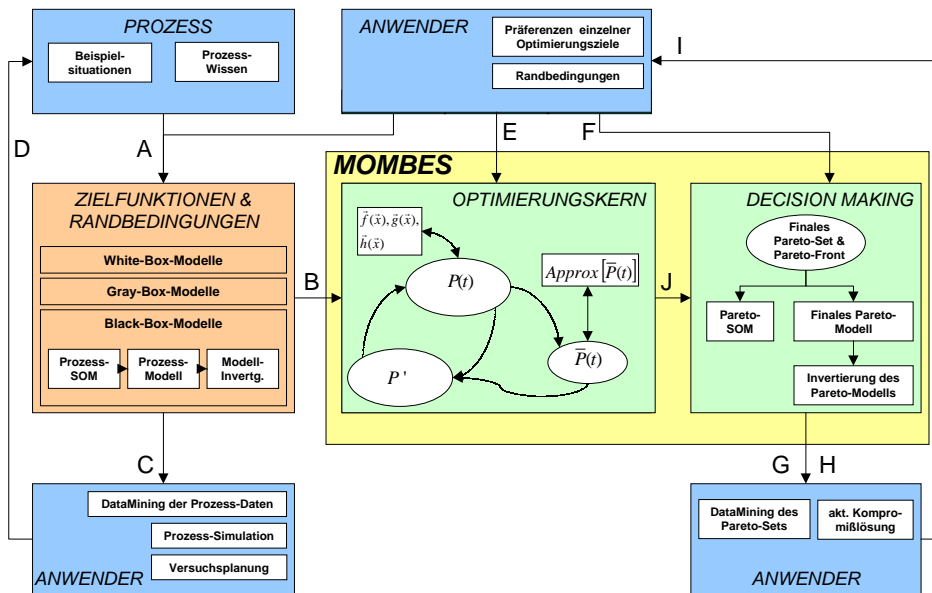


Abbildung 3.1: Einordnung von MOMBES in die modellbasierte Mehrzieloptimierung

Gegeben sei ein Prozeß, der sich durch eine Menge von Tupeln aus Eingangs- und Ausgangsgrößen (Beispielsituationen) oder spezielles Wissen in Form von Regeln und Berechnungsvorschriften über den Zusammenhang zwischen Eingangs- und Ausgangsgrößen darstellen kann (A). In Abhängigkeit davon, in welcher Form das Wissen über die Zusammenhänge zwischen Eingangs- und Ausgangsgrößen vorliegt, kann die Berechnung der Zielfunktionen und Randbedingungen mit Hilfe verschiedenster Modelle (White-Box, Gray-Box oder Black-Modelle) erfolgen. Liegt das Prozeßmodell vor, kann der Anwender flexibler Modellversuche durchführen, als dies am realen Prozeß möglich ist (C). So können Erkenntnisse über das Prozeßverhalten in unbekanntem Betriebspunkten gewonnen und verschiedenste Prozeßsituationen simuliert werden, deren Durchführung in der Realität unter Umständen zu gefährlich oder zu komplex ist. Durch Erkenntnisse, die durch das Prozeßmodell gewonnen werden, können gezielt spezielle Arbeitspunkte des Modells validiert werden (Versuchsplanung) und durch abermaliges Belernen des Modells seine sukzessive Verfeinerung erreicht werden (D).

Im Rahmen der Gesamtlösungsmethodologie liegt die Motivation einer möglichst guten Modellierung von Zielfunktionen und Randbedingungen darin, diese im Verlaufe der evolutionären Suche stabil und mit hoher Modellsicherheit bei der Berechnung der Fitness der Individuen zu verwenden (B).

Das System MOMBES besteht aus zwei Komponenten: dem Optimierungskern und dem Decision-Making-Modul. Durch den Optimierungskern wird eine Menge pareto-optimaler Lösungen generiert auf die der Anwender in der Phase des Decision-Makings zugreifen kann. Dabei können durch das System auch interpolierend Lösungen vorgeschlagen werden, die durch eine Modellierung von P^* auf PF^* gewonnen werden. Zusätzlich können qualitative Zusammenhänge zwischen P^* und PF^* sowie zwischen den Entscheidungsvariablen und Zielfunktionen selbst durch Einsatz einer SOM interaktiv durch den Anwender gewonnen werden.

Im folgenden werden die Module Optimierungskern und Decision-Making-Modul detailliert beschrieben.

3.3 Optimierungskern

Die Optimierung stellt den Hauptteil von MOMBES dar, für den ein spezieller MOEA entwickelt wurde, der in seiner Grundstruktur einem generischen EA entspricht (Algorithmus 1):

INPUT	N : Größe der Grundpopulation; \bar{N} : Größe der Elitepopulation; t_{\max} : maximale Anzahl von Generationen μ : Anzahl Eltern; λ : Anzahl Nachkommen; ρ : Anzahl Wettkämpfer; $\sigma_i^{(0)}$: Initialwert der Mutationsschrittweiten; p_a : Wahrscheinlichkeit für das Einfügen approximierter Individuen
OUTPUT	P^* : Pareto-Set; PF^* : Pareto-Front
1	Initialisierung der Grund- und Elitepopulation: $t = 0$; $P(0) = \emptyset$; $\bar{P}(0) = \emptyset$ <i>for</i> ($i = 0$; $i < N$; $i++$) { wähle $\bar{x}_i \in X$ für eine bestimmte Verteilung; $\bar{a}_i = (\bar{x}_i, \bar{\sigma})$ $P(0) = P(0) + \{\bar{a}_i\}$ }
2	Fitneßberechnung: $\forall \bar{a}_i \in P(t) : \Phi(\bar{a}_i) = FIT(\bar{a}_i, P(t))$
3	Aktualisierung der Elitepopulation $ACT[P(t) \rightarrow \bar{P}(t)]$
4	Berechnung approximierter Individuen \bar{a}^{New} mit einer Wahrscheinlichkeit von p_a und $ACT[\bar{a}^{New} \rightarrow \bar{P}(t)]$ Erzeugen neuer Individuen: $P' = \emptyset$ <i>for</i> ($i = 0$; $i < \lambda$; $i++$) {
5	Selektion von \bar{a}_{E1} und \bar{a}_{E2}
6	Rekombination: $\bar{a}_K = REKOM(\bar{a}_{E1}, \bar{a}_{E2})$
7	Mutation: $\bar{a}_K = MUTAT(\bar{a}_K)$
8	Ergänzen der Zwischenpopulation: $P' = P' + \{\bar{a}_K\}$ }
9	Fitneßberechnung: $\forall \bar{a}_i \in P' : \Phi(\bar{a}_i) = FIT(\bar{a}_i, P')$
10	Übernehme die λ besten Individuen aus P' in $P(t)$
11	$t = t + 1$
12	Wenn $t > t_{\max}$ gehe zu Schritt 13 sonst gehe zu Schritt 3
13	Speichern der Ergebnisse P^* und PF^*
14	Stop

Algorithmus 1: Grundalgorithmus des MOMBES-Optimierungskerns

Für die Berechnung der hierarchischen Fitneß eines Individuums \bar{a}_i mit dem Vektor der Entscheidungsvariablen \bar{x} (Gleichung [4]) wird zunächst für jedes Individuum der aktuellen Population $P(t)$ der Vektor der Zielfunktionen $\vec{f}(\bar{x})$ sowie die Verletzung der Randbedingungen $\vec{g}(\bar{x})$ und $\vec{h}(\bar{x})$ berechnet. Anschließend wird für jedes \bar{a}_i der Pareto-Rang $rank(\bar{a}_i)$ nach Goldberg [Goldberg_1989] bestimmt. Dabei erhalten alle Individuen, die nichtdominierten Lösungen entsprechen, den Rang 1 und werden bei der Bestimmung der weiteren Rangplätze nicht mehr berücksichtigt. Aus der Restpopulation werden wiederum alle nichtdominierten Lösungen bestimmt – sie erhalten den um 1 erhöhten Rang und werden

wiederum bei der Berechnung der Ränge der restlichen Individuen nicht mehr berücksichtigt usw.. Anschließend wird der maximal auftretende Rang bestimmt.

Individuen, die nicht den Randbedingungen entsprechen, erhalten als Fitneßwert die Summe aus dem maximal auftretenden Pareto-Rang sowie dem Wert der Straffunktion $p(\vec{x})$. Hingegen wird Individuen, die Randbedingungen entsprechen, als Fitneßwert direkt der Pareto-Rang zugewiesen.

$$FIT(\vec{a}_i, P) = \begin{cases} rank(\vec{a}_i); & \text{für } \vec{x} \in X_{feasible} \\ \max\{\{rank(\vec{a}_i) \mid \forall \vec{a}_i \in P\}\} + p(\vec{x}); & \text{für } \vec{x} \in X_{feasible} \end{cases} \quad [4]$$

$$\text{mit: } \vec{x} = \Gamma(\vec{a}) \quad \text{und} \quad p(\vec{x}) = \sum_{i=1}^o h_i(\vec{x}) + \sum_{i=1}^m \partial(g_i(\vec{x}))$$

$$\text{sowie } \partial(g_i(\vec{x})) = \begin{cases} -g_i(\vec{x}) & \text{für } g_i(\vec{x}) < 0 \\ 0 & \text{für } g_i(\vec{x}) \geq 0 \end{cases}$$

Es existiert neben der Grundpopulation P von N Individuen eine zweite Elitepopulation \bar{P} , die alle zum aktuellen Zeitpunkt t nichtdominierten Individuen enthält. Die Elitepopulation wird nach jedem Generationszyklus aktualisiert: wird beim Archivieren der dominanten Individuen in \bar{P} eine vorgegebene Anzahl \bar{N} von Elementen überschritten, werden solange diejenigen Individuen aus der Elitepopulation entfernt, die den geringsten euklidischen Abstand zu ihren unmittelbaren Nachbarn auf der Pareto-Front besitzen, bis die Elitepopulation genau \bar{N} Elemente enthält.

Der Optimierungskern ist als MOEA in Form einer Evolutionären Strategie mit Selbstanpassung der Muatationsschrittweiten σ_i für jede Entscheidungsvariable x_i realisiert.

Zur Selektion wird Wettkampfselektion eingesetzt, die auf der Vereinigungsmenge der Grundpopulation und der Elitepopulation aufsetzt. Bei gleicher Fitneß wird dasjenige Individuum selektiert, dessen euklidischer Abstand zum nächsten Nachbarn der Pareto-Front aus \bar{P} größer ist.

Die Entscheidungsvariablen selbst werden nicht binär kodiert. Die Mutation der Individuen erfolgt nach dem Grundschemata einer Standard-ES (Gleichung [5]).

$$\begin{aligned} \sigma_i' &= \sigma_i \cdot \exp(\tau' \cdot N(0,1) + \tau \cdot N_i(0,1)) \\ \tau' &\approx \frac{1}{\sqrt{2 \cdot n}} \quad \tau \approx \frac{1}{\sqrt{2 \cdot \sqrt{n}}} \quad \text{empfohlene Werte} \\ x_i' &= x_i + \sigma_i' \cdot N(0,1) \end{aligned} \quad [5]$$

Bei der Rekombination wird diskrete Rekombination für die Entscheidungsvariablen und intermediäre Rekombination für die Mutationsschrittweiten verwendet. Bei der diskreten Rekombination wird zufällig eine Komponente aus \vec{x} ausgewählt und mit einer bestimmten Wahrscheinlichkeit von einem der Eltern übernommen, wobei die Auswahlwahrscheinlichkeit mit der Fitneß des Eltern-Individuums wächst.

Nach jeder Generationen wird mit einer Wahrscheinlichkeit von p_a ein sogenanntes „approximiertes Individuum“ erzeugt. Dies geschieht, indem zunächst aus den k Zielfunktionen $f_i^p : i = 1, \dots, k; p = 1, \dots, n_{PF^*}$ der n_{PF^*} Elemente der aktuellen Pareto-Front PF^* die Pseudo-Gewichte $\vec{w}^p = [w_1^p, \dots, w_k^p]$ für jeden Punkt p auf der Pareto-Front berechnet wird (Gleichung [6]).

$$w_i^p = \frac{f_i^{\max} - f_i^p}{f_i^{\max} - f_i^{\min}} \text{ mit } \sum_{i=1}^k w_i^p = 1; \forall p \mid p = 1, \dots, n_{PF^*} \leq \bar{N} \text{ und } n_{PF^*} = |PF^*(t)| \quad [6]$$

Anschließend wird mit diesen n_{PF^*} Vektoren \vec{w}^p eine $n_{SOM} \times m_{SOM}$ -SOM belehrt. Die Neuronenanzahl der SOM ist für $k \geq 3$ quadratisch, bei $k = 2$ wird $m_{SOM} = 1$ gewählt. Die Wahl von $m_{SOM} = 1$ bei zweidimensionalen Zielkonflikten ist insofern sinnvoll, da die Abhängigkeit der Gewichte über alle Elemente p der Pareto-Front linear ist und die Kohonenkarte bestrebt ist, dieser linearen Funktion zu folgen (d.h. in der Sammon-Projektion einer 2-dimensionalen Karte die Neuronen bestrebt sind, sich linear anzuordnen). Ab $k \geq 3$ erfolgt eine Projektion in eine $(k - 1)$ -dimensionale Ebene. Die Anzahl der Neurone der Kohonenkarte sollte etwa der maximal zu berechnenden Anzahl der Elemente der Pareto-Front entsprechen d.h. $n_{SOM} \cdot m_{SOM} \approx \bar{N}$. Das Training erfolgt nach dem Algorithmus von Kohonen [Kohonen_2001]. Ist die SOM belehrt, wird der Schwerpunkt des maximalen euklidischen Abstandes der Kopplungsvektoren der SOM zwischen zwei Winner-Neuronen auf der Karte berechnet. Dieser Schwerpunkt entspricht der neuen Kombination von Gewichtungen \vec{w}^{New} mit dem ein schneller gradientenbasierter Optimierungsalgorithmus nach Levenberg/Marquardt (LM) [NumRec_1994] angestoßen wird, indem das MOP durch gewichtete Summenbildung

mittels \vec{w}^{New} in ein SOP überführt wird: $\min \left(\sum_{i=1}^k w_i^{New} \cdot f_i(\vec{x}) \right)$.

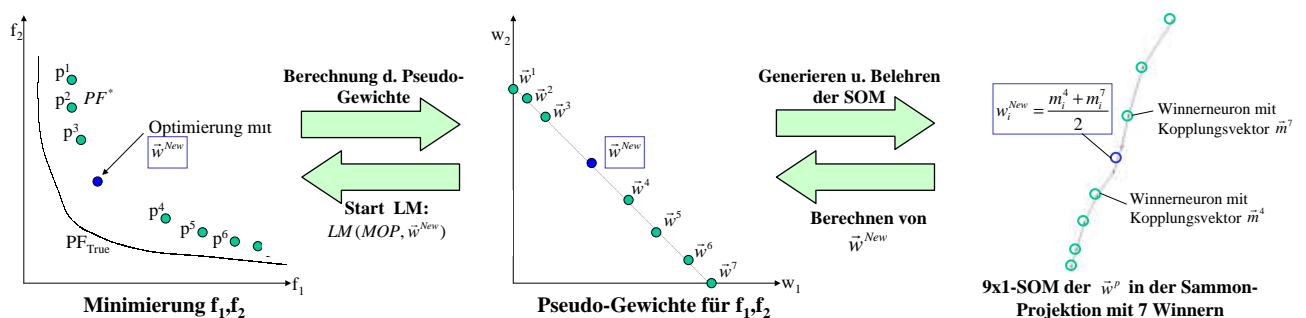


Abbildung 3.2: Schematische Darstellung der Generierung approximierter Individuen

Im Ergebnis des LM, liegen die Entscheidungsvariablen für das approximierte Individuum vor. Sowohl die Startwerte für den LM als auch die Strategieparameter des approximierten Individuums werden von einem zufällig ausgewählten Individuum übernommen, das mit einem der Winnerneurone korrespondiert, das für die Berechnung von \vec{w}^{New} verwendet wurde. Abbildung 3.2 stellt die obenbeschriebene Vorgehensweise nochmals schematisch dar:

Die folgenden Abbildungen illustrieren die Berechnung der neuen Gewichtekombination am Beispiel eines MOP's mit 3 Zielfunktionen:

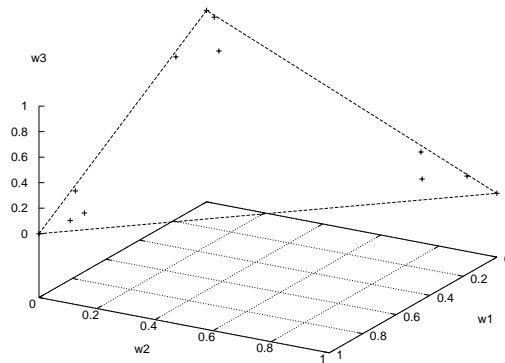


Abbildung 3.3: Verteilung der w_i^P für eine Paretofront mit $|PF| = \bar{N} = 12$

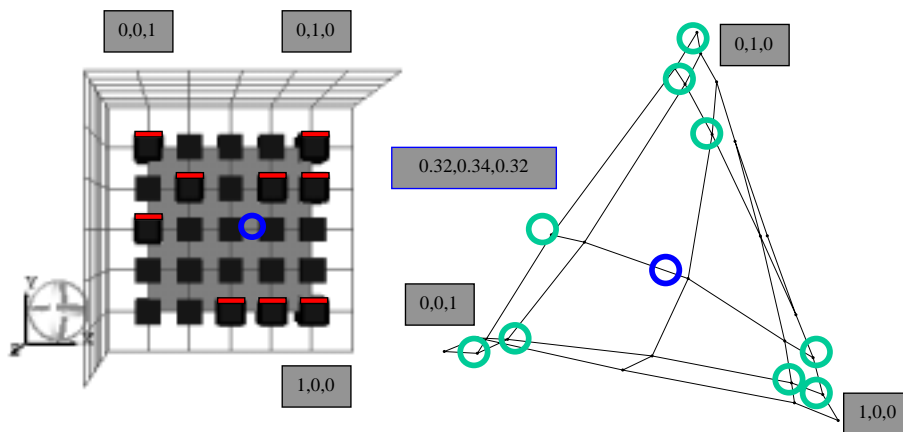


Abbildung 3.4: 5x5-Kohonenkarte mit Winnern und SAMMON-Projektion sowie berechnetem Schwerpunkt

Als Ergebnis der Optimierung entsteht das Pareto-Set P^* und die entsprechende Pareto-Front PF^* von jeweils maximal \bar{N} Elementen, die aus den dominanten Individuen der Elitepopulation gewonnen werden. Dabei bilden die Entscheidungsvariablen der dominanten Individuen das Pareto-Set und die Werte ihrer Zielfunktionen die Pareto-Front.

3.4 Decision-Making-Modul

Das Decision Making ist der zweite Hauptbestandteil von MOMBES. Hier wird dem Benutzer ermöglicht, Wissen über die Zusammenhänge der Elemente der Pareto-Front und des Pareto-Sets sowie deren Elemente untereinander zu extrahieren. Weiter ist es möglich, interaktiv auf pareto-optimale Lösungen zuzugreifen, ohne den Optimierungskern abermals zu durchlaufen. Das Decision-Making-System basiert auf dem durch den Optimierungskern generierten Pareto-Set P^* und er entsprechenden Pareto-Front PF^* (J in Abbildung 3.1).

Zur qualitativen Analyse des Pareto-Sets und der Pareto-Front wird eine SOM belehrt. Auf diese Weise erfolgt eine nichtlineare Projektion des hochdimensionale Datenraums (\vec{x} aus P^* und

\vec{f} aus PF^*) auf eine 2-dimensionale Karte. Die entstandene Karte gestattet Rückschlüsse auf die Zusammenhänge zwischen den Entscheidungsvariablen und Zielfunktionen sowie den Größen untereinander innerhalb der Pareto-Front, da sie Signalähnlichkeiten der Tupel $\{(\vec{x}^1, \vec{f}^1), \dots, (\vec{x}^p, \vec{f}^p)\}$ mit $p = 1, \dots, n_{PF^*}$ in Lagenachbarschaften auf der Kohohnekarte umsetzt. Der Einsatz einer speziellen Visualisierungstechnik ermöglicht die simultane Darstellung der Kopplungsvektoren der SOM in bis zu 11 Dimensionen (siehe Abbildung 3.5)

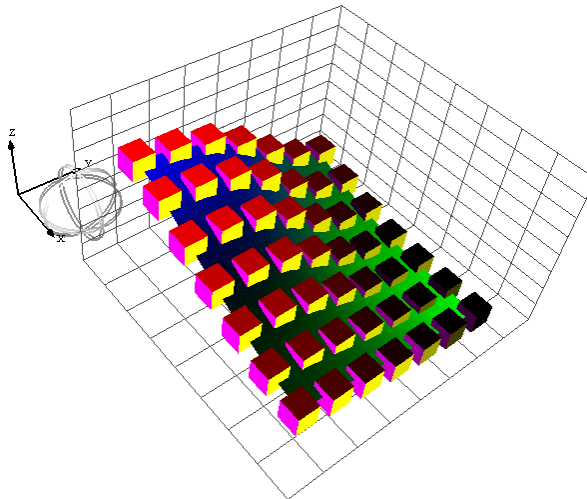


Abbildung 3.5: Visualisierung der Kopplungen einer Pareto-SOM

Eine wichtige Eigenschaft von MOMBES ist die Fähigkeit, des interaktiven und kontinuierlichen Zugriffs auf die approximierte Pareto-Front unter Berücksichtigung von Zielvorgaben und ohne abermaliges Durchlaufen des Optimierungskerns. Dazu wird ein vollvernetztes Feed-Forward-RBF-Netz mit den Vektoren aus P^* (als Input) und den Vektoren aus PF^* (als Zieloutput) nach dem Algorithmus von Poggio/Girosi [PogGi_1989] belehrt. Durch Anwendung des Modells ist eine glatte Approximation der Pareto-Front möglich. Der Anwender kann nun nicht nur auf einzelne Lösungspunkte des Pareto-Sets zugreifen, sondern für beliebige Kombination der einzelnen f_i innerhalb der Pareto-Sets die entsprechenden Lösungen \vec{x} durch invertierte Modellbildung generieren. Zur Invertierung des Modells können Einschränkungen des Wertebereichs für jedes x_i aus \vec{x} , jedes f_i aus \vec{f} sowie der Sicherheit der RBF-Approximation durch Beschränkungen eines Konfidenzmaßes berücksichtigt werden. Die Invertierung selbst ist durch ein schnelles Gradientenabstiegsverfahren nach Levenberg/Marquardt realisiert. Das Konfidenzmaß wird durch die Anwendung eines Maximumsoperators auf die Aktivitäten der hidden-Neurone des RBF-Netzes gebildet und gibt Aufschluß darüber, wie gut ein Datensatz durch die Trainingsvektoren abgesichert ist bzw. kann als Ähnlichkeitsmaß des durch das RBF-Netz verarbeiteten Inputs \vec{x} zu den trainierten Stützstellen des Modells d.h. den Vektoren aus P^* interpretiert werden.

Im Ergebnis des Decision Makings erhält der Anwender eine Kompromißlösung, die seinen speziellen Anforderungen entspricht.

3.5 Abgrenzung von MOMBES zu anderen MOEA

MOMBES stellt einen neuen Ansatz dar, bisherige MOEA auf der Basis eines multi-hybrid-Konzepts um die gezielte Steuerung der Evolution in Richtung homogener verteilter Pareto-Front sowie um ein intelligentes Decision-Making-Modul zu erweitern. Es ist in ein Konzept zur modellbasierten Mehrzieloptimierung in der Prozeßindustrie eingebettet.

Folgende Eigenschaften der oben beschriebenen MOEA wurden für MOMBES übernommen:

- die Anwendung des Prinzips der Pareto-Dominanz als Basis der Fitneßberechnung (SPEA, SPEA2, NSGA, MOGA, PESA, NPGA),
- der Einsatz des Eliteprinzips in Form einer externen Population für dominante Individuen (SPEA, SPEA2, NSGA2),
- die Verwendung des euklidischen Abstandes zwischen Nachbarelementen der Pareto-Front als Kriterium der Aktualisierung der Elitepopulation (PESA),
- die Kombination von MOEA mit lokalen Suchstrategien (Ansatz von D.Quagliarella und A.Vincini) sowie
- der Einsatz von SOM bei EA (SOM-MOEA).

Neu entwickelt wurden dagegen folgende Komponenten:

- das gezielte Einbringen approximierter Individuen während der Optimierung in Bereichen geringer Dichte der Pareto-Front,
- ein interaktives Decision-Making-Modul, das nach Abschluß der Optimierung durch Anwendung eines Interpolationsverfahrens den kontinuierlichen Zugriff auf die Pareto-Front ohne abermaliges Durchlaufen der Optimierung ermöglicht sowie
- die qualitative Analyse der Zusammenhänge zwischen den Vektoren der Pareto-Front und des Pareto-Sets sowie ihrer Komponenten untereinander.

4 Simulationen

Um die Leistungsfähigkeit von MOMBES objektiv einschätzen zu können, wurde der Optimierungskern den für MOEA bekannten Vergleichsfunktionen unterzogen sowie die Güte der Approximation des Decision-Making-Moduls systematisch getestet.

Der Vergleich der Leistungsfähigkeit des Optimierungskerns von MOMBES mit den in Kapitel 2 dargestellten MOEA erfolgt nach dem Vergleich der Güte der generierten Pareto-Front nach den in den Abschnitten 4.1 und 4.2. dargestellten Kriterien.

Um die Interpolationsfähigkeit des Decision-Making-Moduls zu testen, wurde folgende Herangehensweise gewählt: im Anschluß an die Optimierung wird eine Menge $f_i^{Targ et}$ erzeugt, deren Werte gleichverteilt, zufällig über dem durch die PF^* definierten Bereich liegt. Anschließend werden durch invertierte Modellbildung des finalen Pareto-Modells Lösungen $\vec{x}^{Targ et}$ generiert, die bei Propagierung durch das RBF-Netz, den Fehler zwischen dem entsprechenden Netz-Output f_i^{Net} und dem vorgegebenen $f_i^{Targ et}$ minimieren. Anschließend werden mittels $\vec{x}^{Targ et}$ und der für die verwendeten MOP's zur Verfügung stehenden Gleichungen die Werte der Zielfunktionen f_i^{Calc} berechnet. Die prozentuale Abweichung

zwischen $f_i^{Targ et}$ und f_i^{Calc} gibt einen Hinweis auf die Interpolationsfähigkeit des Decision-Making-Moduls.

4.1 Leistungskriterien für MOEA

Zur Einschätzung der Güte der Lösungen eines MOP's - d.h. der Güte seiner Pareto-Front -, existieren verschiedene objektive Leistungskriterien, die im folgenden kurz vorgestellt und in ihren Stärken und Schwächen charakterisiert werden sollen. Dabei wird die Fähigkeit der einzelnen Leistungskriterien beschrieben, eine Pareto-Front hinsichtlich

- der Dominanz ihrer Elemente bezüglich einer Vergleichs-Pareto-Front,
- ihres Abstandes zur theoretisch möglichen Pareto-Front PF_{True} ,
- ihrer Ausbreitung über den Raum der Zielfunktionen und
- ihrer Homogenität

charakterisieren. Während sich die Kriterien 1-4 und 6 auf die Charakterisierung einer einzelnen Pareto-Front beziehen, ist das Kriterium 5 ein Parameter, der das Verhältnis der Leistungsfähigkeit von zwei Pareto-Fronten beschreibt.

1. **Error Ratio:** Der Parameter $ER(PF^*)$ [VeldLa_1999] gibt das prozentuale Verhältnis der Anzahl der Elemente von PF^* an, die in der Menge von PF_{True} enthalten sind (Gleichung [7]).

$$ER(PF^*) = \frac{1}{n_{PF^*}} \cdot \sum_{i=1}^N e_i \quad \text{mit} \quad e_i = \begin{cases} 0 & \text{if } f^i \in PF_{True} \\ 1 & \text{if } f^i \notin PF_{True} \end{cases} \quad [7]$$

$$\text{mit } PF^* = \{f^i \mid i = 1..n_{PF^*}\} \text{ und } n_{PF^*} = |PF^*|$$

Im Sinne der Error Ratio ist eine PF^* dann besonders leistungsfähig, wenn $ER(PF^*)$ möglichst gering ist. Der Einsatz dieser Maßzahl als Vergleichskriterium ist jedoch nur dann sinnvoll, wenn die theoretisch erreichbare Pareto-Front PF_{True} bekannt ist. Es können keine Aussagen über den Grad der Entfernung einer Pareto-Front zu PF_{True} , ihre Dominanz gegenüber einer Vergleichsfront, ihre Ausbreitung im Raum der Zielfunktionen sowie ihre Homogenität gemacht werden. Wird $ER(PF^*)$ beispielsweise zum Vergleich der Pareto-Fronten PF_A^* , PF_B^* und PF_C^* herangezogen (Abbildung 4.1; PF_A^* : grün; PF_B^* : rot; PF_C^* :blau; PF_{True} :schwarz), werden PF_A^* und PF_B^* gleich ($ER(PF_A^*) = ER(PF_B^*) = 100\%$), PF_C^* jedoch als leistungstärkste Pareto-Front mit $ER(PF_C^*) = 0\%$ bewertet. Diese Einschätzung widerspricht jedoch ihrer tatsächlichen Qualität. Danach wäre PF_B^* als am hochwertigsten einzuschätzen, da sie zum einen über die größte Abdeckung im Raum der Zielfunktionen, zum anderen über eine gute Konvergenz d.h. große Nähe der Elemente der Paretofront zu PF_{True} sowie über eine gleichmäßige Verteilung ihrer Elemente verfügt.

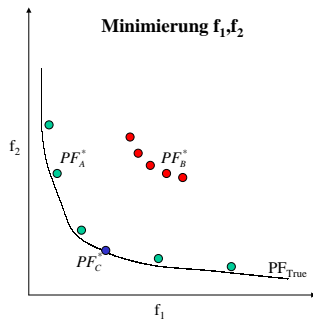


Abbildung 4.1: Beispiel zur Bewertung der Performancemaße ER , GD und Δ

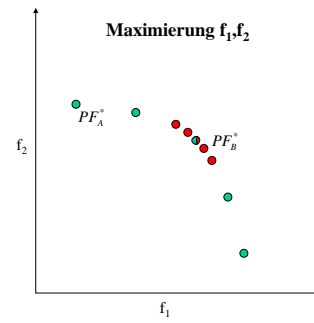


Abbildung 4.2: Beispiel zur Bewertung der Performancemaße C und S

2. **Generational Distance:** Der Parameter $GD(PF^*)$ [VeldLa_1999] entspricht dem Mittelwert der jeweils kleinsten euklidischen Abstände der Elemente von PF^* zu PF_{True} (Gleichung [8]) und ist ebenfalls nur dann als Leistungskriterium einsetzbar, wenn PF_{True} bekannt ist.

$$G(PF^*) = \frac{1}{|PF^*|} \sum_{i=1}^{n_{PF^*}} \min \left(\|f^i - b\| \mid b \in PF_{true} \right) \quad [8]$$

Eine Pareto-Front PF^* wird bezüglich der Generational Distance dann als besonders leistungsfähig eingeschätzt, wenn $GD(PF^*)$ sich dem Wert 0 nähert. Mit Hilfe von $GD(PF^*)$ ist der Abstand einzelner Elemente einer Pareto-Front zu PF_{True} meßbar, nicht jedoch ihre Homogenität und Variabilität oder die Dominanz ihrer Elemente gegenüber einer Vergleichsfront. Als alleiniges Maß zur Charakterisierung einer Pareto-Front ist $GD(PF^*)$ daher ebenso ungeeignet: entsprechend dem in Abbildung 4.1 dargestellten Beispiel gilt: $GD(PF_C^*) < GD(PF_A^*) < GD(PF_B^*)$, was wie oben erläutert nicht den tatsächlichen Eigenschaften der drei Pareto-Fronten entspricht.

3. **Deviation Measure:** Deb charakterisiert in [Deb_2000]⁴ die Homogenität der Verteilung der Elemente einer Pareto-Front bzw. ihre Streuung mit dem Parameter $\Delta(PF^*)$ (Gleichung [9]) mit d_i als euklidischer Distanz zweier benachbarter Elemente der Pareto-Front PF^* und \bar{d} als deren Mittelwert). Eine Pareto-Front ist bezüglich dieses Kriteriums dann besonders leistungsfähig, wenn sich $\Delta(PF^*)$ dem Wert 0 nähert. Als alleiniges Maß für die Güte einer Pareto-Front ist dieser Parameter ebenfalls ungeeignet, da er weder die Abdeckung im Raum der Zielfunktionen, noch die Konvergenz der Zielfunktionen d.h. ihren Abstand zu PF_{True} oder die Dominanz einzelner Elemente gegenüber einer Vergleichsfront berücksichtigt. Entsprechend dem in Abbildung 4.1 dargestellten Beispiel, gilt: $\Delta(PF_A^*) = \Delta(PF_B^*) = \Delta(PF_C^*) = 0$.

⁴ Ähnliche, auf der Streuung der Elemente innerhalb einer Pareto-Front basierende Maßzahlen finden sich in [HornNafpl_1993], [SrinDeb_1994] und [FonsFlem_1998].

$$\Delta(PF^*) = \sum_{i=1}^{n_{PF^*}} \frac{|d_i - \bar{d}|}{n_{PF^*}} \quad [9]$$

4. **Distance to Ideal Vector:** Coello verwendet in [Coello_1996] zur Charakterisierung der Güte eines MOEA u.a. den geringsten, euklidischen Abstand $L_p(PF^*)$ eines Elements der Pareto-Front zum Idealvektor (siehe Definition 4), d.h. den euklidischen Abstand, der sich aus dem Vektor der Zielfunktionen für die best-over-all-Lösung $\vec{f}(\vec{x}^B)$ und dem Idealvektor \vec{f}_{ideal} ergibt. $L_p(PF^*)$ läßt keinerlei Aussagen über die Ausbreitung, die Dominanz, den Abstand zu PF_{True} und die Homogenität von PF^* zu und ist daher lediglich für den Vergleich der Qualität einer einzelnen Lösung, nicht jedoch einer Pareto-Front geeignet.

$$L_p(PF^*) = \left\| \vec{f}_{ideal} - \vec{f}(\vec{x}^B) \right\| \quad [10]$$

5. **Coverage:** Zitzler führt in [ZitzlerTh_1998] einen Parameter $C(PF_1^*, PF_2^*)$ ein, der das Verhältnis der Dominanz der Elemente zweier Pareto-Fronten PF_1^* und PF_2^* charakterisiert: $C(PF_1^*, PF_2^*)$ bestimmt den prozentualen Anteil der Elemente von PF_2^* , die durch mindestens ein Element aus PF_1^* schwach dominiert werden (Gleichung [11]). Ein Wert von $C(PF_1^*, PF_2^*) = 0\%$ bedeutet, daß kein Element aus PF_2^* von Elementen aus PF_1^* , $C(PF_1^*, PF_2^*) = 100\%$ hingegen, daß jedes Element aus PF_2^* von mindestens einem Element aus PF_1^* schwach dominiert wird. Unter Verwendung von $C(PF_1^*, PF_2^*)$ ist es auch ohne die Kenntnis von PF_{True} möglich, das Verhältnis der Güte zweier Pareto-Fronten bezüglich der Pareto-Dominanz ihrer Elemente zu bewerten. Keine Aussagen können jedoch hinsichtlich ihrer geometrischen Ausdehnung im Raum der Zielfunktionen und ihre Homogenität gemacht werden. So gilt für das in Abbildung 4.2 dargestellte Beispiel: $C(PF_A^*, PF_B^*) = C(PF_B^*, PF_A^*) = 0\%$ (PF_A^* :grün; PF_B^* :rot), d.h. beide Pareto-Fronten werden als gleichwertig im Sinne der Coverage eingeschätzt.

$$C(PF_1^*, PF_2^*) := \frac{\left| \left\{ b \in PF_2^* \mid \exists a \in PF_1^* a \preceq b \right\} \right|}{\left| PF_2^* \right|} \quad [11]$$

6. **Size of the Dominated Space:** Ebenfalls in [ZitzlerTh_1998] wird der Parameter S als absoluter, in beschränktem Maße die Ausdehnung einer Pareto-Front beschreibender Parameter, eingeführt. Dabei berechnet sich S aus der Summe der Volumina der Hyperquader, die von den Elementen der Pareto-Front zu einem Referenzpunkt im Raum der Zielfunktionen aufgespannt werden (für Maximierungsproblem ist dieser Referenzpunkt i.d.R. der Koordinatenursprung im Raum der Zielfunktionen). Beim Vergleich der für zwei Pareto-Fronten berechneten Werte für S können keine Schlüsse auf die Dominanz der Elemente der Fronten zueinander gezogen werden. Auch läßt sich nicht der Abstand der untersuchten Pareto-Front zu PF_{True} , und die Homogenität der Verteilung nur beschränkt mit S beschreiben. Besonders ungünstig ist es, daß die Größe von S von der Wahl des

Referenzpunktes abhängt. Gerade für Minimierungsprobleme läßt sich zeigen, daß zwei Pareto-Fronten PF_1^* und PF_2^* existieren, die bei zwei verschiedenen Referenzpunkten I und II folgendes Verhalten besitzen: $S(PF_1^*)^I \neq S(PF_2^*)^{II}$ [Know_2002].

Zusammenfassend wird festgestellt, das für den Vergleich zweier Pareto-Fronten keines der oben aufgelisteten Kriterien in der Lage ist, gleichzeitig die Dominanz der Elemente und ihre geometrische Ausdehnung im Raum der Zielfunktionen zufriedenstellend zu beschreiben. Aus diesem Grunde soll ein neues Leistungskriterium: das **kombiniertes Dominanz-Ausdehnungskriterium (KDA)** eingeführt werden, das dieser Anforderung genügt.

4.2 Einführung des kombinierten Dominanz-Ausdehnungskriteriums (KDA) für MOEA

Gegeben seien zwei Mengen von (Zielfunktions-)Vektoren A und B . Es ist eine Maßzahl zu ermitteln, die sowohl das Verhältnis der Anzahl der dominierten Vektoren gegenüber der jeweiligen Vergleichsmenge, als auch das Verhältnis ihrer geometrische Verteilung über den k -dimensionalen Raum beschreibt, wobei k die Dimension der einzelnen Vektoren sei.

Dazu werden zunächst alle Komponenten a_i mit $\vec{a} \in A$ und b_i mit $\vec{b} \in B$ mit Hilfe der Obergrenzen og_i und Untergrenzen ug_i - die sich in der entsprechenden Dimension i aus der Vereinigungsmenge $A \cup B$ berechnet - auf den Bereich $[0,1]$ normiert. Anschließend werden A und B in die Mengen A' und B' transformiert, indem aus beiden Mengen diejenigen Vektoren entfernt werden, die von mindestens einem Vektor der Vergleichsmenge dominiert werden, d.h. es bleiben alle diejenigen Vektoren in der Menge enthalten, die dominant bezüglich der Elemente der Vereinigungsmenge $A \cup B$ sind (Gleichung [12]).

$$A' = \{ a \in A \mid \neg \exists b \in B : a \succ b \} \quad [12]$$

$$B' = \{ b \in B \mid \neg \exists a \in A : b \succ a \}$$

Es wird nun ein regelmäßiges Hypergitter konstruiert, dessen Ränder von den minimalen (0) und maximalen (1) Werten der Vektorelemente beider Vergleichsmengen in der entsprechenden Dimension i bestimmt sind. Die Anzahl der Gittersegmente n_g in jeder Dimension i wird nach der maximalen Anzahl der Elemente aus A oder B berechnet (Gleichung [13]).

$$n_g = \max(|A|, |B|) \quad [13]$$

$$l = \frac{1}{n_g} \quad [14]$$

Die Seitenlänge l der gleichseitigen Gittersegmente in jeder Dimension i des Gitters, berechnet sich nach Gleichung [14].

Es wird nun für jeweils A' und B' die Anzahl der Gittersegmente gezählt, die mindestens einen Vektor aus A' bzw. B' enthalten. Das Verhältnis dieser beiden Maßzahlen definiert das kombinierte Dominanz-Ausdehnungskriterium, das mit $KDA(A, B)$ bezeichnet wird. Ist die

Menge B' leer, wird $KDA(A, B)$ der Anzahl der Elemente in A' gleichgesetzt. Ist die Menge A' leer, ist $KDA(A, B)$ gleich 0. Es gilt:

$$KDA(A, B) = \frac{1}{KDA(B, A)} \quad [15]$$

Eine Pareto-Front PF_A^* wird gegenüber einer Pareto-Front PF_B^* als leistungsstärker bezüglich des KDA eingeschätzt, wenn $KDA(PF_A^*, PF_B^*) > 1$. Je größer der Wert für $KDA(PF_A^*, PF_B^*)$, desto leistungsfähiger ist PF_A^* gegenüber PF_B^* .

Für das in Abbildung 4.3 dargestellte Beispiel (PF_A^* : grün; PF_B^* : rot) werden nach der oben beschriebenen Vorgehensweise folgende Werte für das KDA ermittelt: $KDA(PF_A^*, PF_B^*) = 5/2 = 2.5$ und $KDA(PF_B^*, PF_A^*) = 2/5 = 0.4$; interpretiert man diese Werte, so ist die Pareto-Front PF_A^* bezüglich der Pareto-Dominanz und geometrischen Ausdehnung um den Faktor 2.5 leistungsfähiger als die die Pareto-Front PF_B^* .

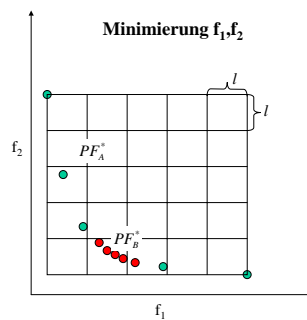


Abbildung 4.3: Berechnung des kombinierten Dominanz-Ausdehnungskriteriums

Für die in Abbildung 4.1 und Abbildung 4.2: illustrierten Beispiele liefert ϖ folgende Ergebnisse:

- Abbildung 4.1: $KDA(PF_A^*, PF_B^*) > KDA(PF_B^*, PF_A^*)$, $KDA(PF_A^*, PF_C^*) > KDA(PF_C^*, PF_A^*)$ und $KDA(PF_C^*, PF_B^*) > KDA(PF_B^*, PF_C^*)$; damit wird nach dem KDA-Kriterium die Pareto-Front PF_A^* als die leistungsstärkste und als leistungsschwächste die Pareto-Front PF_B^* noch vor PF_C^* eingeschätzt.
- Abbildung 4.2: $KDA(PF_A^*, PF_B^*) > KDA(PF_B^*, PF_A^*)$: obwohl alle Elemente in $PF_A^* \cup PF_B^*$ bezüglich der Pareto-Dominanz gleichwertig sind, wird PF_A^* aufgrund der geometrischen Ausdehnung als leistungsstärker als PF_B^* eingeschätzt.

Damit ist für diese Beispiele gezeigt, daß es mit Hilfe des kombinierten Dominanz-Ausdehnungskriteriums möglich ist, die Leistungsfähigkeit zweier Pareto-Fronten bezüglich ihres Dominanz-Verhaltens *und* ihrer geometrischen Ausdehnung zu beschreiben.

In den folgenden Tests werden die verschiedenen MOEA hinsichtlich der Güte der durch sie erzeugten Pareto-Fronten bezüglich der Kriterien KDA , C sowie L_p verglichen. Damit kann ihr

Verhalten bezüglich Konvergenz, Dominanz und homogener Ausbreitung im Raum der Zielfunktionen beschrieben werden.

4.3 Vergleich mit MOEA der ersten Generation

Die in diesem Abschnitt verwendeten Optimierungsprobleme sind den Quellen [Deb_1999] und [Zitzler_1999] entnommen.

Nach Deb [Deb_1999] werden zum Leistungsvergleich Optimierungsprobleme mit zwei Zielfunktionen betrachtet, die der Definition [16].

$$\min \vec{f}(\vec{x}) = [f_1(x_1), f_2(\vec{x})] \quad [16]$$

mit $f_2(\vec{x}) = g(x_2, \dots, x_n) \cdot h(f_1(x_1), g(x_2, \dots, x_n))$ und $\vec{x} = (x_1, \dots, x_n)$

Dabei ist f_1 nur von der Entscheidungsvariable x_1 abhängig - f_2 hingegen eine Funktion der Entscheidungsvariablen ungleich x_1 d.h. $f_2 = f(\vec{x})$ mit $x_i \neq x_1$. Durch die Modifikation von $g(\vec{x})$ und $h(\vec{x})$ ist die Realisierung von Testproblemen möglich, die verschiedenen Problemgruppen genügen.

An dieser Stelle soll auf Probleme mit konvexen (ZTD1), konkaven (ZTD2), diskreten (ZTD3) und multimodalen (ZTD4) Pareto-Fronten eingegangen werden. Wie man leicht sieht, wird die Pareto-Front bei den Testproblemen ZTD1, ZTD2 und ZTD3 bei $g = 1$; für das Testproblem ZTD4 bei $g = 1.25$ herausgebildet. Für das Problem ZTD4 existieren 21^9 lokale Pareto-Fronten – es ist daher besonders geeignet, die Fähigkeit multimodale Optimierungsprobleme zu lösen, zu demonstrieren [Zitzler_1999].

ZTD1	$f_1(x_1) = x_1, \quad g(x_2, \dots, x_n) = 1 + 9 \cdot \frac{\sum_{i=2}^n x_i}{n-1}, \quad h(f_1, g) = 1 - \sqrt{\frac{f_1}{g}},$ <p>mit $n = 30$ und $x_i \in [0,1]$</p>	[17]
ZTD2	$f_1(x_1) = x_1, \quad g(x_2, \dots, x_n) = 1 + 9 \cdot \frac{\sum_{i=2}^n x_i}{n-1}, \quad h(f_1, g) = 1 - \left(\frac{f_1}{g}\right)^2,$ <p>mit $n = 30$ und $x_i \in [0,1]$</p>	[18]
ZTD3	$f_1(x_1) = x_1, \quad g(x_2, \dots, x_n) = 1 + 9 \cdot \frac{\sum_{i=2}^n x_i}{n-1}, \quad h(f_1, g) = 1 - \sqrt{\frac{f_1}{g}} - \frac{f_1}{g} \cdot \sin(10 \cdot \Pi \cdot f_1),$ <p>mit $n = 30$ und $x_i \in [0,1]$</p>	[19]
ZTD4	$f_1(x_1) = x_1, \quad g(x_2, \dots, x_n) = 1 + 10 \cdot (n-1) + \sum_{i=2}^n (x_i^2 - 10 \cdot \cos(4 \cdot \Pi \cdot x_i)),$	[20]

	$h(f_1, g) = 1 - \sqrt{\frac{f_1}{g}}, \text{ mit } n = 10 \text{ und } x_1 \in [0,1] \text{ sowie } x_i \in [-5,+5], i = 2, \dots, n$	
--	--	--

Neben einer reinen Zufallssuche (RAND) wurden folgende MOEA für die Test nach [16] bis [20] verwendet: VEGA, HPGA, NPGA, NSGA, MOGA, SPEA. Die Ergebnisse der genannten MOEA für die Testprobleme wurden von E. Zitzler auf seiner Homepage (<http://www.tik.ee.ethz.ch/~zitzler/>) allen Interessenten zugänglich gemacht und auch für die Vergleiche innerhalb dieser Arbeit verwendet.

Folgende Parametereinstellungen gelten für die oben genannten Algorithmen: Anzahl der Kodierungsbits pro Entscheidungsvariable $n_{Bits} = 30$, Anzahl der Generationen: $T = 250$, Anzahl der Individuen: $N = 100$, 1-Punkt-Cross-Over mit einer Wahrscheinlichkeit von $p_c = 0.8$, Mutation pro Bit mit einer Wahrscheinlichkeit von $p_m = 0.01$, Nischenradius: $\sigma_{share} = 0.4886$, Größe der Vergleichsmenge beim NPGA: $t_{dom} = 10\%$ und Größe der Populationen bei SPEA: $N = 80$ und $\bar{N} = |\bar{P}_t| = 20$.

Für MOMBES wurden folgende Parametereinstellungen vorgenommen: $\mu = 15$, $\lambda = 7$, $\rho = 3$, Größe der externen Elite-Population: $\bar{N} = |\bar{P}_t| = 20$, Wahrscheinlichkeit für das Einfügen approximierter Individuen $p_a = 0.02$. Um einen fairen Vergleich zwischen MOMBES und den zitierten MOEA durchzuführen, wurde MOMBES nach einer festen Anzahl von Funktionsaufrufen der Zielfunktion, die den oben genannten Parametereinstellungen der Genetischen Algorithmen entsprechen, abgebrochen: $n_{function\ call} = 100 + (2 \cdot 100 \cdot 250) = 50\ 100$. Die Initialisierung der Startpopulation erfolgte durch gleichverteilte zufällige Belegung der \vec{x} über ihren Wertebereich.

Für jeden MOEA wurden 30 Testläufe durchgeführt. Für den Vergleich der MOEA untereinander wurde diejenige Pareto-Front mit dem größten KDA-Wert aus diesen Testläufen ausgewählt.

Für die Auswertung wurden die Zielfunktionswerte von jeweils maximal 100 dominanten Individuen verwendet, die nach Abschluß der Optimierung in der aktuellen Population und der Elitepopulation zur Verfügung standen.

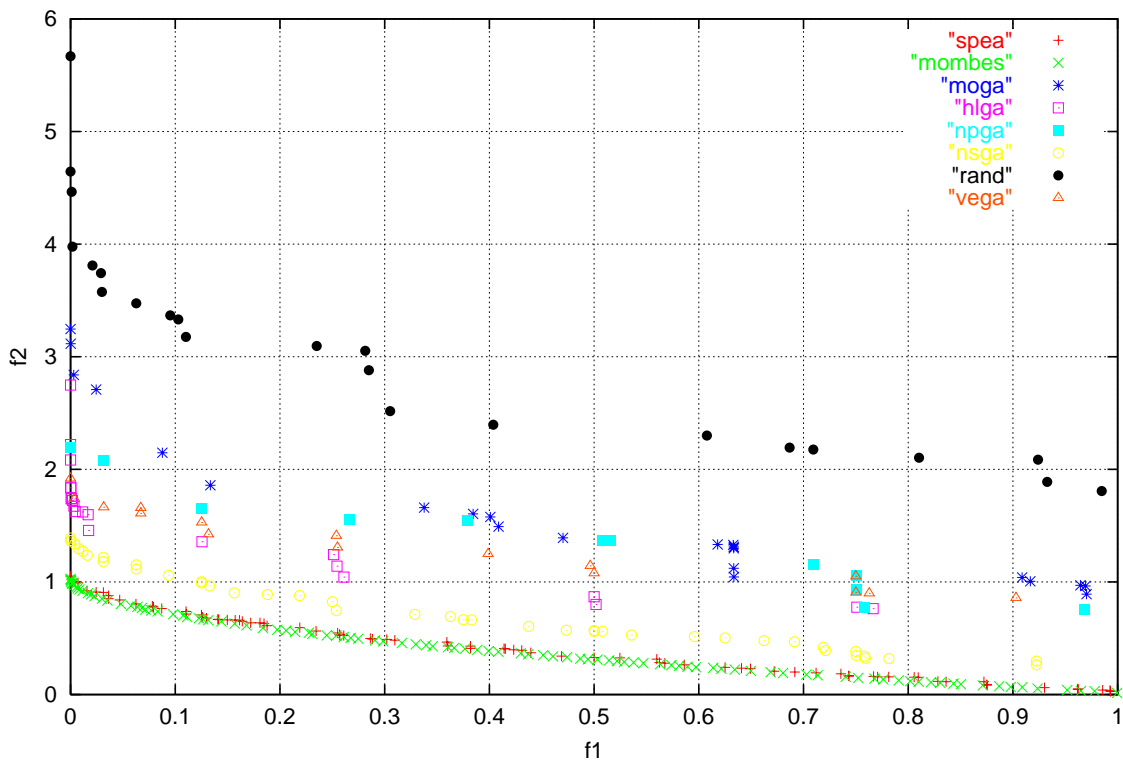


Abbildung 4.4: Pareto-Fronten der verschiedenen MOEA für ZDT1

KDA(A,B)		B								
A		RAND	NPGA	MOGA	VEGA	HLGA	NSGA	SPEA	MOMBES	Mittelwert
	RAND		0	0	0	0	0	0	0	0
	NPGA	14,00		1,50	0,15	0,25	0,00	0,00	0,00	2,27
	MOGA	23,00	0,67		0,08	0,05	0,02	0,00	0,00	3,40
	VEGA	17,00	6,50	13,00		0,09	0,00	0,00	0,00	5,23
	HLGA	22,00	4,00	22,00	10,99		0,00	0,00	0,00	8,43
	NSGA	44,00	44,00	44,00	44,00	44,00		0,00	0,00	31,43
	SPEA	81,00	81,00	81,00	81,00	81,00	81,00		0,028	69,43
	MOMBES	100,00	100,00	100,00	100,00	100,00	100,00	35,0		92,71

Tabelle 4.1: $KDA(A, B)$ für ZDT1

i	n_{Tests}	$\max \ f_i^{Targ et} - f_i^{Calc}\ $	$\min \ f_i^{Targ et} - f_i^{Calc}\ $	$mean \ f_i^{Targ et} - f_i^{Calc}\ $
1	1000	14,05%	0%	6,73%
2	1000	20,58%	0%	7,24%

Tabelle 4.2: Interpolationsfähigkeit von MOMBES für ZDT1

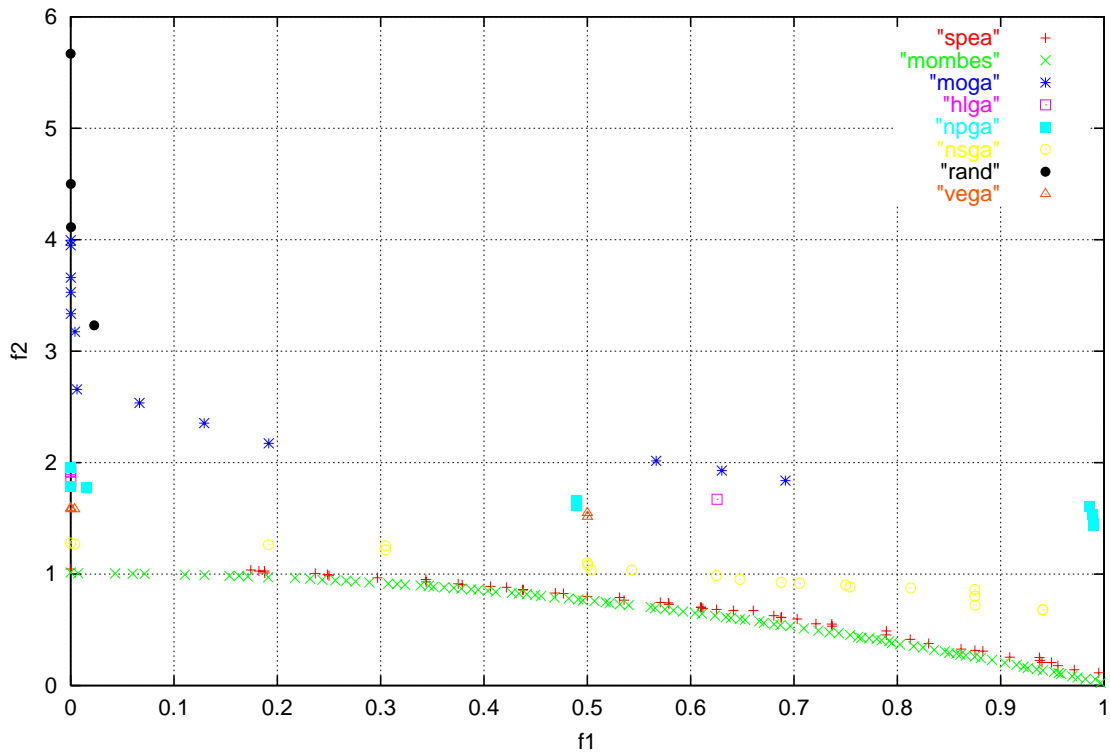


Abbildung 4.5: Pareto-Fronten der verschiedenen MOEA für ZDT2

KDA(A,B)		B								
		RAND	HLGA	MOGA	VEGA	NPGA	NSGA	SPEA	MOMBES	Mittelwert
A	RAND		0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	HLGA	5,00		5,00	0,00	0,33	0,00	0,00	0,00	1,48
	MOGA	14,00	0,00		0,00	0,00	0,00	0,00	0,00	2,00
	VEGA	6,00	6,00	6,00		3,00	0,00	0,00	0,00	3,00
	NPGA	10,00	3,00	10,00	0,33		0,00	0,00	0,00	3,33
	NSGA	22,00	22,00	22,00	22,00	22,00		0,00	0,00	15,71
	SPEA	54,00	54,00	54,00	54,00	54,00	54,00		0,00	46,29
	MOMBES	100,00	100,00	100,00	100,00	100,00	100,00	100,00		100,00

Tabelle 4.3: $KDA(A, B)$ für ZDT2

i	n_{Tests}	$\max \ f_i^{Target} - f_i^{Calc}\ $	$\min \ f_i^{Target} - f_i^{Calc}\ $	$mean \ f_i^{Target} - f_i^{Calc}\ $
1	1000	13,29%	0%	3,81%
2	1000	16,75%	0%	3,94%

Tabelle 4.4: Interpolationsfähigkeit von MOMBES für ZDT2

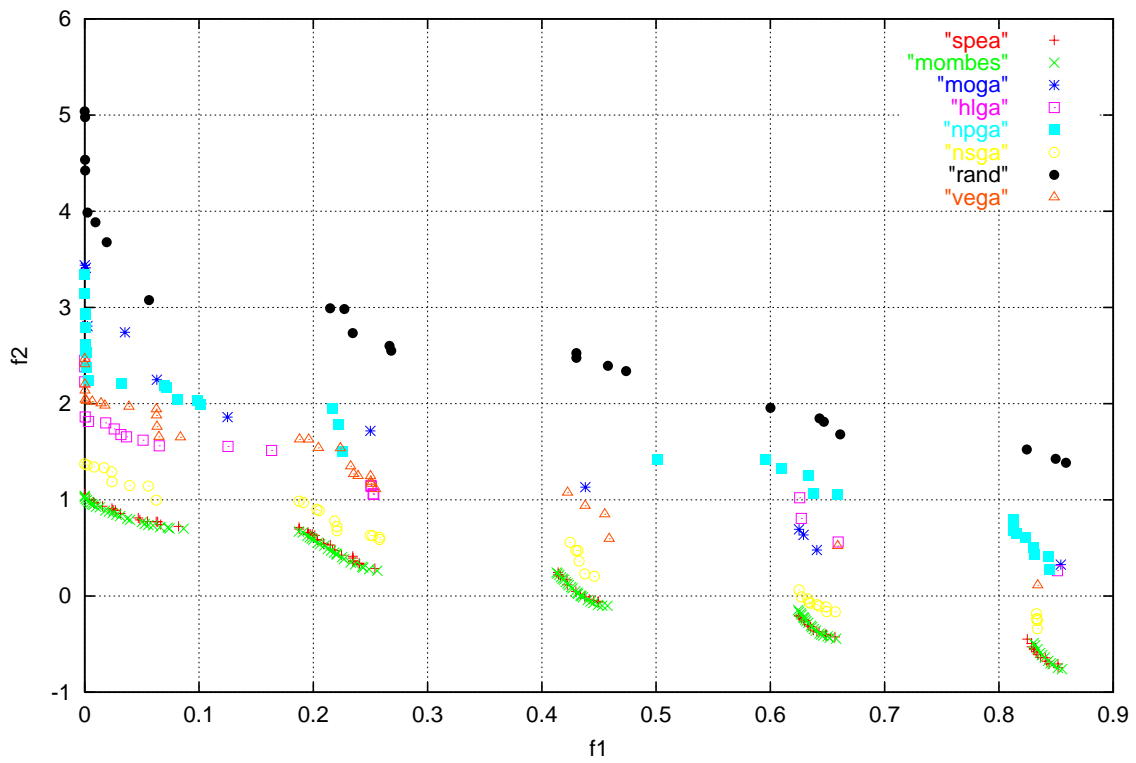


Abbildung 4.6: Pareto-Fronten der verschiedenen MOEA für ZDT3

KDA(A,B)		B								
A		RAND	MOGA	HLGA	NPGA	VEGA	NSGA	SPEA	MOMBES	Mittelwert
	RAND		0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	MOGA	13,00		0,15	0,31	0,04	0,00	0,00	0,00	1,93
	HLGA	22,00	6,50		4,00	1,09	0,00	0,00	0,00	4,80
	NPGA	34,00	3,20	0,25		0,12	0,00	0,00	0,00	5,37
	VEGA	33,00	23,00	0,92	8,33		0,00	0,00	0,00	9,32
	NSGA	42,00	42,00	42,00	42,00	42,00		0,00	0,00	30,00
	SPEA	85,00	85,00	85,00	85,00	85,00	85,00		0,577	72,93
	MOMBES	100,00	100,00	100,00	100,00	100,00	100,00	1,73		85,96

Tabelle 4.5: $KDA(A, B)$ für ZDT3

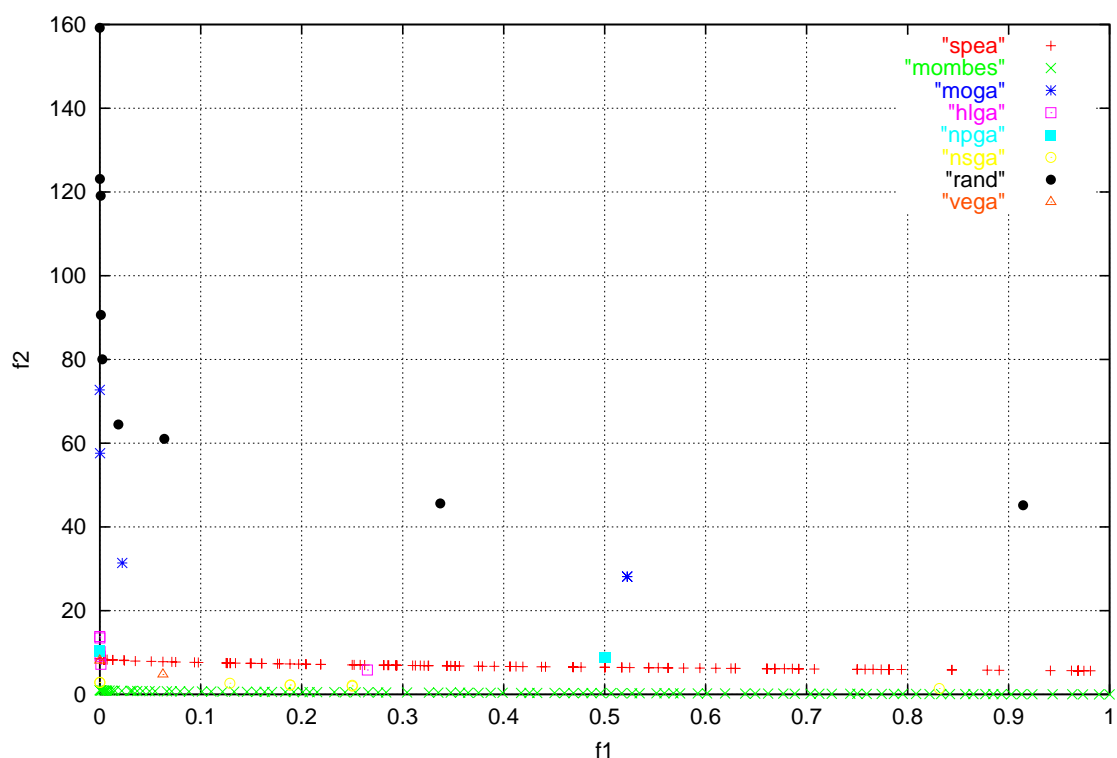


Abbildung 4.7: Pareto-Fronten der verschiedenen MOEA für ZDT4

KDA(A,B)		B								
		RAND	MOGA	HLGA	NPGA	VEGA	NSGA	SPEA	MOMBES	Mittelwert
A	RAND		0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	MOGA	6,00		0,00	0,00	0,00	0,00	0,00	0,00	0,86
	NPGA	5,00	5,00	0,50		0,00	0,00	0,00	0,00	1,50
	VEGA	3,00	3,00	2,00	3,00		0,00	1,0	0,00	1,71
	HLGA	7,00	7,00		2,00	0,50	0,00	0,22	0,00	2,38
	NSGA	9,00	9,00	9,00	9,00	9,00		9,00	0,00	7,71
	SPEA	100,00	100,00	4,5	100,00	1,0	0,00		0,00	43,64
	MOMBES	100,00	100,00	100,00	100,00	100,00	100,00	100,00	100,00	100,00

Tabelle 4.6: Performancevergleich bezüglich $KDA(A, B)$ für ZDT4

i	n_{Tests}	$\max \ f_i^{Target} - f_i^{Calc}\ $	$\min \ f_i^{Target} - f_i^{Calc}\ $	$mean \ f_i^{Target} - f_i^{Calc}\ $
1	1000	7,92%	0%	4,36%
2	1000	8,27%	0%	5,66%

Tabelle 4.7: Interpolationsfähigkeit von MOMBES für ZDT4

4.4 Vergleich mit MOEA der zweiten Generation

Während im Abschnitt 4.3 MOMBES mit den MOEA der ersten Generation d.h. HLGA, MOGA, VEGA, NPGA, NSGA und SPEA verglichen wurde, soll nun der Vergleich mit moderneren MOEA wie NSGA2, SPEA2 und PESA erfolgen. Als Testproblem wurde ein sphärisches MOP mit 3 Zielfunktionen betrachtet, das aus [Schaffer_1985] entnommen wurde und sich u.a. auch in [ZitzLauTh_2001] und [LauRuSch_2001] findet.

$$\text{SPH-3: } \min \vec{f}(\vec{x}) = [f_1(\vec{x}), f_2(\vec{x}), f_3(\vec{x})] \quad f_j(\vec{x}) = \sum_{i=1, i \neq j}^n x_i^2 + (x_j - 1)^2; 1 \leq j \leq 3 \quad [21]$$

$$\text{mit } n = 100 \text{ und } x_i \in [-10^3, +10^3]; i = 1, \dots, n$$

Auch für dieses Problem konnten die Ergebnisse der betrachteten MOEA der oben erwähnten Homepage von E. Zitzler entnommen werden.

Alle Entscheidungsvariablen der betrachteten MOEA wurden nicht binärcodiert, sondern als Fließkommazahlen dargestellt. Der Operator für Mutation und Rekombination wurde mittels des SBX-20 Operator realisiert [ZitzLauTh_2001]. Die Größe der Population und der maximal zu speichernden dominanten Lösungen wurde wie oben auf 100 festgesetzt. Zur besseren Beurteilung der Leistungsfähigkeit der einzelnen MOEA wurde als Abbruchkriterium die Anzahl der Aufrufe der Zielfunktion mit $n_{\text{function call}} = 100000$, $n_{\text{function call}} = 500000$ und $n_{\text{function call}} = 1000000$ gewählt. Die Parametrierung der MOEA folgt der für die in Abschnitt 4.3 dargestellten MOEA.

Auch hier wurden für jeden MOEA 30 Testläufe durchgeführt. Für den Vergleich der MOEA untereinander wurde jeweils diejenige Pareto-Front mit dem größten KDA-Wert ausgewählt.

Das Konvergenzverhalten wurde mit Hilfe des Abstandes $L_p(PF^*)$ zum Idealvektor (Mittelwert, Streuung und bester Wert über 30 Testläufe) und die Qualität der Pareto-Front mittels KDA verglichen.

MOEA	100.000 function calls			500.000 function calls			1.000.000 function calls		
	\bar{L}_p	σ_{L_p}	L_{PBEST}	\bar{L}_p	σ_{L_p}	L_{PBEST}	\bar{L}_p	σ_{L_p}	L_{PBEST}
SPEA2	5094.10	1131.33	3350,6	3.36	0.35	2,70	1.71	0.09	1,59
NSGA2	4948.34	953.14	2973,6	3.48	0.43	2,72	1.62	0.10	1,51
PESA	72.47	16.52	43,6	2.98	0.35	2,18	1.93	0.11	1,79
MOMBES	6.66	14.20	1,2	1.40	0.11	1,19	1.34	0.09	1,18

Tabelle 4.8: Performancevergleich $L_p(PF^*)$ für SPH-3

KDA(A,B)	B					
		SPEA2	NSGA2	PESA	MOMBES	Mittelwert
A	SPEA2		0,00	0,00	0,00	0,00
	NSGA2	2,00		0,00	0,00	0,67
	PESA	2,00	2,00		0,00	1,33
	MOMBES	100,0	100,0	100,0		100,0

Tabelle 4.9: $KDA(A, B)$ für SPH-3 nach 100.000 function calls

KDA(A,B)	B					
A		SPEA2	NSGA2	PESA	MOMBES	Mittelwert
	SPEA2		0,98	0,00	0,00	0,33
	NSGA2	1,03		0,02	0,00	0,35
	PESA	100	48,50		0,00	49,50
	MOMBES	100	100	100		100

Tabelle 4.10: $KDA(A, B)$ für SPH-3 nach 500.000 function calls

KDA(A,B)	B					
A		SPEA2	NSGA2	PESA	MOMBES	Mittelwert
	SPEA2		0,39	100	0,01	33,47
	NSGA2	2,55		49,00	0,06	17,20
	PESA	0,00	0,02		0,00	0,01
	MOMBES	97,00	16,33	100		71,11

Tabelle 4.11: $KDA(A, B)$ für SPH-3 nach 1.000.000 function calls

i	n_{Tests}	$\max \ f_i^{Target} - f_i^{Calc}\ $	$\min \ f_i^{Target} - f_i^{Calc}\ $	$mean \ f_i^{Target} - f_i^{Calc}\ $
1	1000	24,55%	0%	9,62%
2	1000	29,74%	0%	8,22%
3	1000	34,95%	0%	8,56%

Tabelle 4.12: Interpolationsfähigkeit von MOMBES für SPH-3

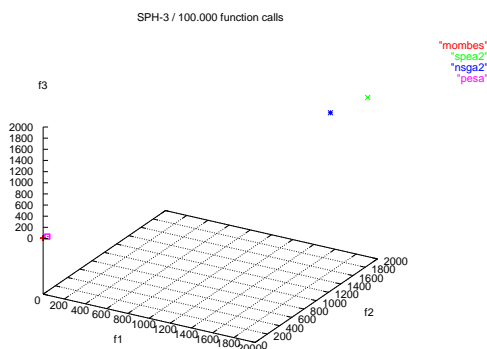


Abbildung 4.8: Pareto-Fronten für SPH-3 nach 100.000 function calls

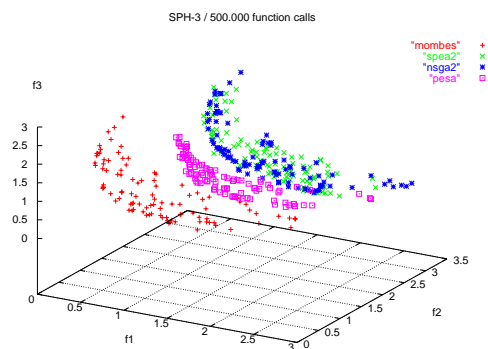


Abbildung 4.9: Pareto-Fronten für SPH-3 nach 500.000 function calls

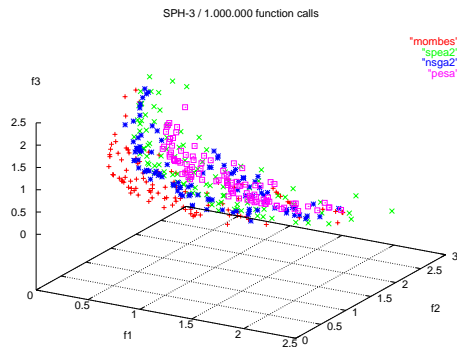


Abbildung 4.10: Pareto-Fronten für SPH-3 nach 1.000.000 function calls

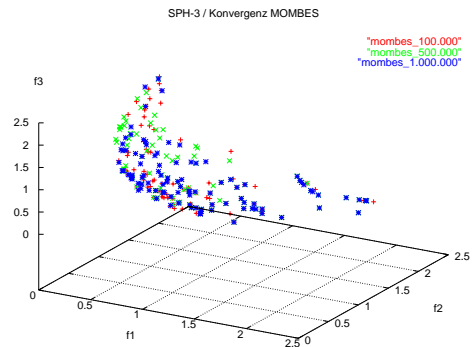


Abbildung 4.11: Pareto-Fronten für SPH-3 für MOMBES nach 100.000, 500.000 und 1.000.000 function calls

4.5 Ergebnisse

Aufgrund der durchgeführten Tests können folgende Aussagen getroffen werden: der entwickelte Algorithmus MOMBES konnte an den Testfunktionen ZDT1-ZDT4 sowie SPH-3 Pareto-Fronten generieren, die in ihrer Qualität hochwertiger als alle durch die betrachteten Konkurrenzalgorithmen generierten Pareto-Fronten sind. Das betrifft sowohl die Konvergenz der Pareto-Fronten als auch ihre Homogenität und Ausbreitung im Raum der Zielfunktionen.

Bei den Simulationen auf Basis der Testfunktionen ZDT1-ZDT4 sind die erzielten Ergebnisse von MOMBES mit denen von SPEA vergleichbar. Alle von MOMBES und SPEA generierten Lösungen dominieren bei ZDT1-ZDT3 alle durch RAND, NPGA, MOGA, VEGA, HLGA und NSGA generierten Lösungen d.h. der Wert für $KDA(SPEA,*)$ und $KDA(MOMBES,*)$ entspricht der Größe der durch SPEA und MOMBES generierten Pareto-Fronten: $KDA(SPEA,*) = |PF_{SPEA}^*|$ und $KDA(MOMBES,*) = |PF_{MOMBES}^*|$. Damit sind beide Algorithmen diesen MOEA für diese Testfunktionen klar überlegen. Bei ZDT1 ist das Verhältnis der Lösungen von MOMBES, die die Lösungen von SPEA stark dominieren und gleichmäßig verteilt sind ca. 35:1. Das bedeutet, daß die 35-fache Menge an stark dominanten und entsprechend dem KDA gleichmäßig verteilten Lösungen gegenüber SPEA generiert werden konnten. Bei ZDT2 und ZDT4 dominieren alle Lösungen von MOMBES die von SPEA stark, bei ZDT3 hingegen ist die Leistungsfähigkeit beider Algorithmen etwa gleichwertig: MOMBES liegt nur um den Faktor 2.16 über der Leistungsfähigkeit von SPEA bezüglich KDA. Dieses Verhalten bei SPEA und MOMBES ist hauptsächlich auf den Einsatz des Eliteprinzips in Form der externen Population, auf die Verwendung des Pareto-Rangs als Basis der Fitneßberechnung, die Archivierungsfunktion der externen Elitepopulation sowie auf das Einfügen approximierter Individuen bei MOMBES zurückzuführen. Die für eine ES typische Selbstanpassung der Mutationsschrittweiten bei MOMBES führen zu sehr guten Konvergenzeigenschaften. Sowohl SPEA als auch MOMBES können sehr gut sowohl konvexe als auch konkave Pareto-Fronten generieren. Für die diskontinuierliche Testfunktion ZDT3 geht der Vorteil des Einfügens approximierter Individuen jedoch verloren, da MOMBES in Regionen geringer Dichte der Pareto-Front zwar eine lokale Optimierung anstößt, die jedoch zu keiner Verbesserung der Qualität von PF_{MOMBES}^* führen kann, da die Pareto-Front genau in diesen Bereichen nicht

existiert. Seine Überlegenheit zeigt MOMBES anhand der Testfunktion ZDT4 deutlich: die Pareto-Fronten jedes betrachteten MOEA werden vollständig durch die Elemente der PF_{MOMBES}^* stark dominiert. Erwähnenswert ist weiterhin, das bei allen 4 Testfunktionen jeweils 100 Elemente der PF_{MOMBES}^* generiert werden konnten, während die Anzahl der Elemente der Pareto-Front bei den betrachteten Konkurrenz-MOEA deutlich geringer war: für ZDT1: $|PF_{NPGA}^*| = 14$, $|PF_{MOGA}^*| = 23$, $|PF_{VEGA}^*| = 17$, $|PF_{HLGA}^*| = 22$, $|PF_{NSGA}^*| = 44$ und $|PF_{SPEA}^*| = 81$; für ZDT2: $|PF_{NPGA}^*| = 10$, $|PF_{MOGA}^*| = 14$, $|PF_{VEGA}^*| = 6$, $|PF_{HLGA}^*| = 5$, $|PF_{NSGA}^*| = 22$ und $|PF_{SPEA}^*| = 54$; für ZDT3: $|PF_{NPGA}^*| = 34$, $|PF_{MOGA}^*| = 23$, $|PF_{VEGA}^*| = 33$, $|PF_{HLGA}^*| = 22$, $|PF_{NSGA}^*| = 42$ und $|PF_{SPEA}^*| = 85$ und für ZDT4: $|PF_{NPGA}^*| = 5$, $|PF_{MOGA}^*| = 6$, $|PF_{VEGA}^*| = 3$, $|PF_{HLGA}^*| = 7$, $|PF_{NSGA}^*| = 9$ und $|PF_{SPEA}^*| = 100$. Dieses ist ein deutliches Indiz dafür, das MOMBES in der Lage ist eine große Variation der Population zu ermöglichen. Bezüglich der Dominanz und Ausbreitung der Pareto-Front ist MOMBES allen MOEA der ersten Generation deutlich überlegen.

Die Auswertung der Tests bezüglich der Güte der Approximation durch das Decision-Making-Modul zeigen, daß MOMBES in der Lage ist, ein Modell $P^* \rightarrow PF^*$ zu erzeugen und kontinuierlich auf die approximierte Pareto-Front zuzugreifen. Die mittlere Abweichung zwischen approximierten und berechneten Zielfunktionswerte liegt bei den Testfunktionen ZDT1, ZDT2 und ZDT4 bei nur ca. 5%. Die Approximation für ZDT3 wurde aufgrund der Diskontinuität der Zielfunktionen aus den Betrachtungen ausgenommen.

Ein ähnliches Verhalten wie bei Tests ZDT1-ZDT4 zeigte MOMBES auch im Vergleich mit den MOEA der zweiten Generation auf Basis der Testfunktion SPH-3. Besonders deutlich konnte hier die schnelle Konvergenz des Ansatzes gezeigt werden. Sowohl die Werte für KDA als auch für L_p lagen nach 100.000 und 500.000 Funktionsaufrufen deutlich günstiger als für die betrachteten MOEA: SPEA2, NSGA2 und PESA. Dabei waren nicht nur die Werte für L_p selbst geringer, sondern auch ihre Streuung. Erst nach etwa 1.000.000 Funktionsaufrufen konnten die betrachteten MOEA ähnliche Konvergenzeigenschaften aufweisen. Die sehr schnelle Konvergenz von MOMBES ist bei diesem Beispiel vor allem durch den Einsatz des gradientenbasierten lokalen Suchoperators beim Einfügen approximierter Individuen begründet. Weitere Tests müssen an für Gradientenverfahren weniger sensitiven MOP's durchgeführt werden, um diese Ergebnisse zu bestätigen. Trotzdem kann an dieser Stelle bemerkt werden, daß sich MOMBES auch gegenüber den MOEA der zweiten Generation als überaus leistungsfähig erwiesen hat. Bereits nach 100.000 Zielfunktionsaufrufen lagen 100 Elemente von PF_{MOMBES}^* vor, wogegen durch SPEA2 und NSGA2 nur 2 Elemente der Pareto-Fronten generiert werden konnten. Eine ebenso sehr schnelle Konvergenz konnte bei PESA festgestellt werden: bereits nach 500.000 Zielfunktionsaufrufen lagen 100 Elemente der PF_{PESA}^* vor, die fast alle Elemente von PF_{NSGA2}^* und PF_{SPEA2}^* streng dominierten, jedoch ihrerseits vollständig von den Elementen von PF_{MOMBES}^* streng dominiert wurden. Nach 1.000.000 Zielfunktionsaufrufen lag die Qualität der PF_{PESA}^* jedoch unter der von PF_{SPEA2}^* und PF_{NSGA2}^* . Der KDA-Wert für MOMBES erreichte auch nach Abschluß der Optimierung einen Wert von fast 100 gegenüber SPEA2 und PESA d.h. fast 100% der Elemente von PF_{MOMBES}^* dominierten streng die Elemente von PF_{PESA}^* und PF_{SPEA2}^* . Der NSGA2-Algorithmus arbeitete leistungsfähiger als PESA und SPEA2, war jedoch MOMBES um den Faktor 16,33 unterlegen.

Auch für dieses Beispiel gelang die Modellierung $P^* \rightarrow PF^*$, jedoch mit einem etwas größerem Approximationsfehler als bei ZDT1, ZDT2 und ZDT3. Die Begründung hierfür liegt darin, daß bei 3 Zielfunktionen die Dichte der 100 Trainingsvektoren weniger groß war als bei nur 2 Zielfunktionen und gleicher Anzahl Trainingsvektoren.

Durch das gezielten Einbringens approximierter Individuen ist eine deutliche Beschleunigung der Optimierung und Verbesserung der Qualität der generierten Pareto-Front beobachtbar, d.h. die Fortschrittsgeschwindigkeit konnte drastisch erhöht werden. Es konnte festgestellt werden, daß gerade bei schwierigen multikriteriellen Problemen die gezielte Steuerung der Evolution in Richtung homogener Pareto-Front durch den hier beschriebenen Ansatz vorteilhaft ist. Diese Beschleunigung läßt sich alleine durch eine zufällige Neubildung eines Individuums oder Rekombination zweier dominanter Individuen pro Generation nicht erklären, sondern ist Resultat der gezielten Approximation. Es wurde weiterhin gezeigt, daß die Modellierung der Pareto-Lösungen auf die Pareto-Front möglich ist und mit einer befriedigenden Qualität realisiert werden konnte.

5 Zusammenfassung

Es wurde ein System zur Modellbasierten Mehrzieloptimierung beschrieben, das sich durch die gezielte Steuerung der Optimierung in Richtung homogen verteilter und gut konvergierter Pareto-Front und deren Modellierung während des Decision-Makings auszeichnet. Der beschriebene Ansatz weist folgende Vorteile auf: Der Benutzer

- wird nicht gezwungen, vor dem Optimierungslauf eine tatsächlich in der Praxis schwer zu realisierende numerische Zielgewichtung vorzunehmen, sondern
- kann die Zielvorgaben auch nach der Optimierung variieren, d.h. mehrere Szenarien simulieren. Die interaktive Rolle des Benutzers wird hervorgehoben und die Zielvorgaben werden unscharf formulierbar.
- Während der evolutionären Suche lernt der Algorithmus durch den Einsatz Neuronaler Netze die Evolutionsrichtung und gibt damit der evolutionären Suche eine Richtungsunterstützung, die zusätzlich durch den Einsatz eines gradientenbasierten Optimierungsalgorithmus beschleunigt wird.

Berücksichtigt man, daß Zielkonflikte mit mehr als 3 Dimensionen i.d.R. für den Anwender schwer vorstellbar sind und durch die kombinatorische Explosion der unterschiedlichen Zielgewichtungen eine korrekte Wahl des optimalen „Lösungskorridors“ fast unmöglich ist, wird der Vorteil, erst nach Abschluß der Optimierung auf die Lösungsmenge zuzugreifen, deutlich. So wird eine zu starke a-priori Einschränkung des Lösungsraumes vermieden und statt dessen eine „Wolke“ pareto-optimaler Lösungen generiert, die nach Abschluß der Optimierung modelliert und interpoliert wird.

6 Literatur

- [Bäck_1997] Bäck, T., Fogel, D. and Michalewicz, Z. (Hrsg.) Handbook of Evolutionary Computation, Oxford University Press, Oxford 1997
- [BietNiss_1995] Biethahn, J.; Nissen, V. (Hrsg.) Evolutionary Algorithms in Management Applications; Berlin Springer 1995
- [Broek_1995] Broekmeulen, R.A.C.M.: Facility Management of Distribution Centres of for Vegetables and Fruits; in [BietNiss_1995] S. 199-210
- [BüMiKou_2002] Büche, D.; Milano, M.; Koumoutsakos, P.: Self-Organizing Maps for Multi-Objective Optimization, in Alwyn M. Barry (Hrsg.), GECCO 2002: Proceedings of the Bird of a Feather Workshops, Genetic and Evolutionary Computation Conference, S. 152-155, New York, July 2002
- [Coello_1996] Carlos, C.A.C.: An Empirical Study of Evolutionary Techniques for Multiobjective Optimization in Engineering Design. PhD thesis, Department of Computer Science, Tulane University, New Orleans, LA, April 1996
- [CorKnoOa_2000] Corne, D.W.; Knowles, J.D.; Oates, M.J.: The Pareto Envelope-based Selection Algorithm for Multiobjective Optimization, In Schoenauer, M.; Deb, K.; Rudolph, G.; Xin Yao, Lutton, E.; Merelo J.J.; Schwefel, H.P. (Hrsg.), *Proceedings of the Parallel Problem Solving from Nature VI Conference*, pages 839--848. Springer, 2000
- [De_Jong_1975] De Jong, K.A.: An Analysis of the Behavior of a Class of Genetic Adaptive Systems, Ph.D. Thesis, University of Michigan; 1975
- [Deb_1999] Deb, K.: Multi-objective Genetic Algorithms: Problem Difficulties and Construction of Test Problems. *Evolutionary Computation* 7(3): 205-230 (1999)
- [Deb_2000] Deb, K.; Agrawal, S.; Pratab, A.; Meyarivan, T.: A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II, KanGAL report 200001, Indian Institute of Technology, Kanpur, India, 2000.
- [Deb_2002] Deb, K.: Multi-Objective Optimization using Evolutionary Algorithms; John Wiley & Sons, Chichester 2002
- [DebGoel_2001] Deb, K; Goel, T.: A Hybrid Multi-Objective Evolutionary Approach to Engineering Shape Design. In: [EMO_2001], S. 385-399.
- [EMO_2001] Zitzler, E.; Deb, K.; Thiele, L.; Carlos, C.A.C.; Corne, D. (Hrsg.), First International Conference on Evolutionary Multi-Criterion Optimization, Springer-Verlag. Lecture Notes in Computer Science No. 1993, 2001
- [FonsFlem_1993] Fonseca, C.M.; Fleming P.J.: Genetic Algorithm for multiobjective optimization: Formulation, discussion and generalization. In: [Forrest_1993] S. 416-423
- [FonsFlem_1995] Fonseca, C.M.; Fleming, P.J.; An Overview of Evolutionary Algorithms in Multiobjective Optimization: in *Evolutionary Computation* 3(1), 1995 MIT
- [FonsFlem_1998] Fonseca, C.M.; Fleming, P.J.: Multiobjective Optimization and multiple Constraint Handling with Evolutionary Algorithms – Part I: A unified Formulation; *IEEE Transactions on Systems, Man and Cybernetics – Part A: Systems and Humans*, 28(1): S. 26-37
- [Forrest_1993] Forrest, S. (Hrsg.) (1993), Proceedings of the 5th International Conference on Genetic Algorithms, University of Illinois, Urbana-Champaign IL, July 17-21, 1993, Morgan Kaufmann, San Mateo CA
- [Fourman_1985] Fourman, M.: Compaction of Symbolic Layout using Genetic Algorithms; in [Grefenst_1985] S.141-153
- [Goldberg_1989] Goldberg, D.E.: Genetic algorithms in search, optimization, and machine

- learning, Addison-Wesley, Reading MA; 1989
- [GoldRich_1987] Goldberg, D.E.; Richardson, J.: Genetic Algorithm with sharing for multimodal function optimization. In: Genetic Algorithms and their Applications: Proceedings of the 2. International Conference on Genetic Algorithm, pg. 41-49, Hillsdale, NJ, 1987
- [GomJimIlb_1998] Gomez-Skarmeta, A.F.; Jimenez, F.; Ibanez, J.: Pareto-optimality in fuzzy modeling. In: 6th European Congress on Intelligent Techniques and Soft Computing; S. 694-700; 1998
- [Grefenst_1984] Grefenstette, J.J. :GENESIS: A System for using Genetic Search Procedures; in: Proceedings of the Conference on Intelligent Systems and Machines, 1984; S. 161-164
- [Grefenst_1985] Grefenstette, J.J. (Hrsg): Genetic Algorithms and Their Applications: Proceedings of the 1. International Conference on Genetic Algorithms, Lawrence Erlbaum; 1985
- [Gueugn_1997] Gueugniaud, P.Y.; Bertin-Maghit, M.; Hirschauer, C.; Bouchard, C.; Vilasco, B.; Petit, P.; Gen, M.; Ida, K.; Lee, J.; Kim, J.: Fuzzy nonlinear goal programming using genetic algorithm. Computers and industrial engineering 33(1); S.39-42, 1997
- [HajelaLin_1992] Hajela, P.; Lin, C.-Y.: Genetic search strategies in multicriterion optimal design; Structural Optimization, 4: S.. 99-107
- [Hansen_1997] Hansen, M.P.: Tabu search in multiobjective optimization: MOTS; in: 13th International Conference on Multi-Criterion Decision Making (MCDM'97); University of Cape Town; 1997
- [Herdy_1992] Herdy, M. (1992) Reproductive isolation as strategy parameter in hierarchical organized evolution strategies, in: [MänMan_1992], S. 207-217
- [HornNafpl_1993] Horn, J.; Nafpliotis, N.: Multiobjective optimization using the niched pareto genetic algorithm. IlliGAL Report 93005, University of Illinois at Urbana-Champaign, Urbana, Illinois, USA
- [IshMur_2001] Ishibuchi, H.; Murata, T.: Minimizing the fuzzy rule and maximizing its performance by a multi-objective genetic algorithm. In: [EMO_2001] S. 259-264; 2001
- [Know_2002] Knowles, J.D.: Local-Search and Hybrid Evolutionary Algorithms for Pareto Optimization; Ph.D.Thesis, University of Reading, Department of Computer Science, Reading UK 2002
- [Know_2002] Knowles, J.D.: Local-Search and Hybrid Evolutionary Algorithms for Pareto Optimization; Ph.D.Thesis, University of Reading, Department of Computer Science, Reading UK 2002
- [KnowCorn_2000] Knowles, J.D.; Corne, D. W.: Approximating the nondominated front using the Pareto archived evolution strategy. Evolutionary Computation, 8(2):149—172, 2000.
- [KoepRudl_1998] Köppen, M; Rudlof, S.: Multiobjective Optimization by Nussy Algorithm, In R. Roy, T. Furuhashi, and P.K. Chawdhry, editors, *Advances in Soft Computing*, S. 357-368, London, 1998. Springer.
- [Kohonen_2001] Kohonen, T.: Self-organizing maps. Springer series in information sciences, 3rd ed.; 2001
- [Kursawe_1991] Kursawe, F.: A Variant of Evolutionary Strategies for Vector Optimization; in: [SchwMän_1991]; 1991
- [LauRuSch_2001] Laumanns, M.; Rudolph, G.; Schwefel, H.-P.: Mutation control and convergence in evolutionary multi-objective optimization. In: proceedings of the 7th International Mendel Conference of Soft Computing; (MENDEL 2001), Brno, Czech Republic 2001
- [MänMan_1992] Männer, R., B. Manderick (Hrsg.) (1992) Parallel problem solving from

- nature 2, Proceedings of the 2nd PPSN Conference, Brüssel, 1992
- [Meyer_2000] Meyer, D.: Modellbasierte Mehrzieloptimierung verfahrens-technischer Produktionsprozesse mit Neuronalen Netzen und Evolutionsstrategien, in VDI Berichte 1526, Computational Intelligence im industriellen Einsatz, Tagungsband zur Tagung im Mai 2000 in Baden-Baden
- [MilSchKou_2001] Milano, M.; Schmidhuber, J.; Koumotsakos, P.: Active Learning with Adaptive Grids; International Conference on Artificial Neural Networks, Wien 2001
- [NumRec_1994] Press, W.H.; Teukolsky, S.A.; Vetterling, W.T.; Flannery B.P.: Numerical Recipes in C; 2. Auflage; Cambridge Univ. Press 1994
- [PogGi_1989] Poggio, T., Girosi, F.: A Theory of Networks for Approximation and Learning, A.I. Memo, No. 1140, MIT Artificial Intelligence Laboratory, 1989.
- [QuagliaVic_1997] Quagliarella, D.; Vicini, A.: Coupling genetic algorithms and gradient based optimization techniques. In: Genetic Algorithms and Evolutionary Strategy in Engineering and Computer Science – Recent advances and industrial applications; S. 289-309; Wiley; Chichester 1997
- [Schaffer_1984] Schaffer, J.D.: Some experiments in Maschine Learning using Vector Evaluated Genetic Algorithms, Ph. Thesis, Vanderbilt University, Nashville 1984
- [Schaffer_1985] Schaffer, J.D.: Multible objective optimization with vector evaluated genetic algorithms; in: [Grefenst_1985] S. 93-100
- [Schwefel_1995] H.-P. Schwefel, Evolution and Optimum Seeking; Wiley and son; 1995
- [SchwMän_1991] Schwefel, H.P.; Männer, R. (Hrsg.): Parallel Problem Solving from Nature; LNCS 496, Springer Berlin 1991
- [SrinDeb_1994] Srinivas N., Deb, K.: Multiobjective Optimization using Nondominated Sorting in Genetic Algorithms; in: Evolutionary Computation Vol. 2; S. 221-248; 1994
- [VeldLa_1999] Veldhuizen, D.A.; Lamont, G.B.: Multiobjective Evolutionary Algorithm Test Suits In Janice Carroll, Hisham Haddad, Dave Oppenheim, Barrett Bryant, and Gary B. Lamont, editors, Proceedings of the 1999 ACM Symposium on Applied Computing, pages 351-357, San Antonio, Texas, 1999. ACM.
- [ZitzLauTh_2001] Zitzler, E.; Laumanns, M.; Thiele, L.: SPEA2: Improving the Strength Pareto Evolutionary Algorithm, Technical Report 103, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH) Zurich, Gloriastrasse 35, CH-8092 Zurich, Switzerland, May 2001
- [Zitzler_1999] Zitzler, E.: Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications; Dissertation an der Eidgenössischen Technischen Hochschule Zürich; TIK-Schriftenreihe Nr. 30; Swiss Federal Institut of Technology 1999
- [ZitzlerTh_1998] Zitzler, E.; Thiele, L.: An Evolutionary Algorithm for Multiobjective Optimization: The Strength Pareto Approach; TIK-Report Nr. 43; Swiss Federal Institut of Technology (ETH) 1998
- [ZitzlerTh_1999] Zitzler, E.; Thiele, L.: Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach, IEEE Transactions on Evolutionary Computation, 3(4):257-271, November 1999.