

Stefan Fenn

Vergleich und Erweiterung von globalen Optimierungsverfahren auf nichtglatten Funktionen

Diplomarbeit

BEI GRIN MACHT SICH IHR WISSEN BEZAHLT



- Wir veröffentlichen Ihre Hausarbeit, Bachelor- und Masterarbeit
- Ihr eigenes eBook und Buch - weltweit in allen wichtigen Shops
- Verdienen Sie an jedem Verkauf

Jetzt bei www.GRIN.com hochladen
und kostenlos publizieren



Bibliografische Information der Deutschen Nationalbibliothek:

Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de/> abrufbar.

Dieses Werk sowie alle darin enthaltenen einzelnen Beiträge und Abbildungen sind urheberrechtlich geschützt. Jede Verwertung, die nicht ausdrücklich vom Urheberrechtsschutz zugelassen ist, bedarf der vorherigen Zustimmung des Verlanges. Das gilt insbesondere für Vervielfältigungen, Bearbeitungen, Übersetzungen, Mikroverfilmungen, Auswertungen durch Datenbanken und für die Einspeicherung und Verarbeitung in elektronische Systeme. Alle Rechte, auch die des auszugsweisen Nachdrucks, der fotomechanischen Wiedergabe (einschließlich Mikrokopie) sowie der Auswertung durch Datenbanken oder ähnliche Einrichtungen, vorbehalten.

Impressum:

Copyright © 2008 GRIN Verlag
ISBN: 9783668635067

Dieses Buch bei GRIN:

<https://www.grin.com/document/412495>

Stefan Fenn

Vergleich und Erweiterung von globalen Optimierungsverfahren auf nichtglatten Funktionen

GRIN - Your knowledge has value

Der GRIN Verlag publiziert seit 1998 wissenschaftliche Arbeiten von Studenten, Hochschullehrern und anderen Akademikern als eBook und gedrucktes Buch. Die Verlagswebsite www.grin.com ist die ideale Plattform zur Veröffentlichung von Hausarbeiten, Abschlussarbeiten, wissenschaftlichen Aufsätzen, Dissertationen und Fachbüchern.

Besuchen Sie uns im Internet:

<http://www.grin.com/>

<http://www.facebook.com/grincom>

http://www.twitter.com/grin_com

Diplomarbeit
im Studiengang Diplom-Mathematik

**Vergleich und Erweiterung von globalen
Optimierungsverfahren auf nichtglatten
Funktionen**

Inhaltsverzeichnis

1	Einleitung	4
2	Die Anwendung OptimizeTest	5
2.1	Bedienung	5
2.2	Implementierung	6
3	Testfunktionen für Optimierungsverfahren	7
3.1	Funktionen einer Veränderlichen	7
3.1.1	Differenzierbare Funktionen	7
3.1.2	Stetige Funktionen	8
3.1.3	Unstetige Funktionen	9
3.2	Funktionen mehrerer Veränderlichen	11
3.2.1	Differenzierbare Funktionen	11
3.2.2	Stetige Funktionen	16
3.2.3	Unstetige Funktionen	18
3.2.4	Minigolf-Problem (Abb. 3.28 und 3.29)	19
4	Evolution	23
4.1	Genetik	23
4.1.1	Von der DNA zur RNA	23
4.1.2	Proteine	24
4.1.3	Befruchtung	24
4.1.4	Genetische Variabilität	26
4.2	Vererbung	28
4.3	Artentrennung	30
4.3.1	Präzygotische Fortpflanzungsbarrieren	30
4.3.2	Postzygotische Fortpflanzungsbarrieren	31
4.4	Evolutionenstfaktoren	31
4.4.1	Variabilität innerhalb der Population	31
4.4.2	Inzucht	34
4.5	Evolution als Optimierungsprozess	35
5	Evolutionäre Algorithmen	36
5.1	Evolutionenststrategien und Genetische Algorithmen	37
5.1.1	Prinzipielle Struktur	38

5.1.2	Repräsentation der Individuen	38
5.1.3	Initialisierung	38
5.1.4	Bewertung	38
5.1.5	Fitness	39
5.1.6	Selektion	44
5.1.7	Rekombination	47
5.1.8	Mutation	49
5.1.9	Wiedereinfügen	52
5.1.10	Abbruchkriterien	54
5.2	Mutation-Selektion-Verfahren	55
6	Weitere Optimierungsverfahren	57
6.1	Partikelschwarm-Optimierung	57
6.2	Zyklisches Abstiegsverfahren	59
6.3	STEP	60
6.3.1	Schwierigkeit einer Partition	60
6.3.2	Der STEP-Algorithmus	62
6.4	DIRECT	63
6.4.1	Initialisierung	64
6.4.2	Unterteilung eines Hyper-Rechtecks	64
6.4.3	Auswahl der zu unterteilenden Hyper-Rechtecke	65
6.5	Simulated Annealing	68
6.6	Nelder-Mead-Verfahren	70
6.6.1	Konstruktionsprinzipien	70
6.6.2	Ablauf des Nelder-Mead-Verfahrens	73
6.7	Rosenbrock-Methode	74
6.7.1	Verbessertes Orthogonalisierungsverfahren	75
7	Erweiterung der Algorithmen	78
7.1	Erweiterung bei der Evolutionsstrategie	78
7.1.1	Parallelogramm-Rekombination	78
7.1.2	Elliptische Rekombination	79
7.1.3	Fitnessberechnung mit verminderter Überbevölkerung	79
7.1.4	Rekombination aus der Nachbarschaft	80
7.1.5	Mutation mit Partitionen	82
7.2	Erweiterung von DIRECT	84
7.2.1	DIRECT mit Nelder-Mead	84
7.2.2	DIRECT mit Beschränkung der Partitionsanzahl	86
8	Vergleich der Optimierungsverfahren	88
8.1	Auswahl der Testfunktionen	88
8.2	Auswahl der Parameter	89
8.3	Vergleich der lokalen Optimierungsverfahren	97

8.4	Vergleich der globalen Optimierungsverfahren	97
Anhang		100
A.1	Programmcode: Minigolf-Problem	100
A.2	Programmcode: Methode von Hooke und Jeewes	102
A.3	Programmcode: STEP-Algorithmus	104
A.4	Programmcode: DIRECT-Algorithmus	106
A.5	Programmcode: Nelder-Mead-Verfahren	110
A.6	Programmcode: Rosenbrock-Methode	113
A.7	Programmcode: Mutation mit Partitionen	116

1 Einleitung

In dieser Diplomarbeit werden globale Optimierungsverfahren untersucht, die auf nichtglatten und insbesondere nichtstetigen Funktionen operieren können. Als ich vor vier Jahren an der Fachhochschule Augsburg eine Diplomarbeit über selbstlernende Bewegungsabläufe von vierbeinigen Roboterskelette¹ schrieb, stieg mein Interesse für diese Optimierungsverfahren. Damals habe ich eine sehr einfache Evolutionsstrategie (siehe Abschnitt 5.1) verwendet und die Hauptarbeit bestand in der Art der Kodierung des Bewegungsablaufes.

Unter anderen blieben damals folgende Fragen ungeklärt:

- Kann die Evolutionsstrategie derart gestaltet werden, dass sich die Population bei mehreren lokalen bzw. globalen Optima trennt? Normalerweise steuert im Verlauf der Optimierung die Population nur auf einen dieser Punkte zu.
- Gibt es weitere globale Optimierungsverfahren für nichtglatte Funktionen?

Die Fragen motivierten mich für diese Arbeit. In Kapitel 2 wird die speziell für die Diplomarbeit entwickelte Testumgebung **OptimizeTest** vorgestellt, mit der eine Vielzahl von Optimierungsverfahren getestet werden können. In Kapitel 3 werden die Testfunktionen behandelt, die in **OptimizeTest** enthalten sind. Die Testfunktionen werden zum Vergleichen und Analysieren der Optimierungsverfahren benutzt. Um zu erkennen, welche Prinzipien der biologischen Evolution bei den Evolutionären Algorithmen übernommen werden, wird in Kapitel 4 eine Einführung in die biologische Evolution gegeben. Die Evolutionären Algorithmen insbesondere Evolutionsstrategien sind in Kapitel 4 detailliert beschrieben. Es wurden aber auch andere Optimierungsverfahren gesucht, die in Kapitel 6 beschrieben sind. Im Rahmen meiner Analyse habe ich Modifikationen an den Optimierungsverfahren vorgenommen und diese sind in Kapitel 7 zu finden. Einige Modifikationen beantworten die oben gestellten Fragen. Zuletzt werden in Kapitel 8 die Optimierungsverfahren miteinander verglichen.

¹Ein Beispiel ist unter <http://www.metholution.de/stefan/diplomarbeit/simulation.mpeg> zu sehen.

2 Die Anwendung OptimizeTest

2.1 Bedienung

Das Programm ist im Verzeichnis `optimizeTest` auf der CD zu finden und wird mit `start.bat` gestartet¹. Im oberen Bereich werden die Funktionen und Optimierungsverfahren visualisiert. Im linken unteren Bereich liegt der Navigationsbaum, der beim Anwählen Detail-Informationen im rechten Bereich anzeigt.

Elemente im Navigationsbaum:

- **Optimierungsprobleme:** Hier sind alle Testfunktionen zu finden. Bevor ein Optimierer ausgewählt wird, muss ein Optimierungsproblem selektiert worden sein.
- **Optimierer:** Dieser Ordner enthält alle Optimierungsverfahren, wobei die Parameter nach dem nach der Auswahl des Optimierers unter „Aktuelles Projekt“ gesetzt werden können.
- **Projekte:** Es sind vollständig konfigurierte Szenarien wählbar.
 - **Layouts:** Verschiedene Darstellungen die in Größe und Anzahl der Zeichenflächen variieren.
 - **Optimierer:** Die einzelnen Optimierungsverfahren sind schon einer Testfunktion zugewiesen und die Parameter sind spezifisch eingestellt.
 - **Optimierer/optimierte Parameter:** Es sind die optimalen Parameter eingestellt, die in Abschnitt 8.2 bestimmt wurden.
 - **Vergleich:** Es können Optimierungsverfahren mit einander verglichen werden. Dabei wird darauf geachtet, dass im Mittel beide Optimierer die gleiche Anzahl an Funktionsauswertungen verwenden.
 - **Minigolf:** Für die Minigolf-Simulation wird eine spezielle Darstellung gewählt.
- **Aktuelles Projekt:** Die wichtigsten Einstellungen sind folgende.
 - **Optimierungsaufgabe/Parameter:** Falls das Optimierungsproblem Parameter besitzt, können diese hier eingestellt werden.
 - **<Zeichenflächenname>/Parameter:** Parameter des Optimierungsverfahrens. Wenn man mit der Maus kurz über den Parameternamen bleibt, erscheint eine Kurzinformation.
 - **<Zeichenflächenname>/Schnittstellen:** Die verschiedenen Schnittstellen des Optimierungsverfahren können hier ausgetauscht werden.

¹Die Anwendung läuft schneller, wenn sich das Programmverzeichnis auf der Festplatte befindet.

2.2 Implementierung

OptimizeTest in in Java implementiert und benutzt zur Datenverwaltung XML-Dateien². In diesen XML-Dateien befindet sich auch der Programmcode für die verschiedenen Optimierungsverfahren, die in JavaScript implementiert wurden. Durch JavaScript können, ohne neu kompilieren zu müssen, einfach neue Optimierungsverfahren hinzugefügt werden, indem eine neue XML-Datei geschrieben wird. Ein Nachteil von JavaScript ist die langsame Laufzeit im Vergleich zu Java. Jeder Programmcode, der in dieser Diplomarbeit für Erläuterungen von Algorithmen angegeben ist, wurde in JavaScript geschrieben. Die XML-Dateien werden beim Laden auf Gültigkeit bezüglich eines XML-Schemas überprüft und mit JAXB geladen. Der JavaScript-Code wird mit der Scripting Engine von Java ausgeführt.

Die Darstellung der Funktionen wurde mit der freundlichen Genehmigung von Prof. Dr. Max Weiß aus dem Projekt Math4u2 (www.math4u2.de) übernommen, in dem ich seit sieben Jahren mitarbeite. In Math4u2 habe ich die 2D-Zeichenfläche erweitert und die Höhenkarte vollständig selbst entwickelt.

OptimizeTest hat einen größeren Umfang angenommen und enthält inzwischen 127 Klassen und 790 Methoden. Der Source-Code ist auf der CD im Verzeichnis `optimizeTest/src` zu finden.

²Dies ist ein standardisiertes Format um beliebige Informationen zu beschreiben.

3 Testfunktionen für Optimierungsverfahren

Um die Qualität von globalen Optimierungsverfahren zu bestimmen, werden diese auf verschiedene Testfunktionen angewandt. Bei einigen Testfunktionen, die ausgewählt wurden, ist es sehr schwierig das globale Optimum zu erreichen und bei anderen Funktionen ist die Konvergenzgeschwindigkeit von Interesse. Im Folgenden werden alle benutzten Testfunktionen vorgestellt, wobei auf die Charakteristik der Funktion eingegangen wird.

Alle Funktionen, die vorgestellt werden, können auf die Form

$$\min f(x) \quad \text{unter } l \leq x \leq u \quad \text{mit } l \in \mathbb{R}^N, x \in \mathbb{R}^N, u \in \mathbb{R}^N, f: \mathbb{R}^N \rightarrow \mathbb{R}, N \in \mathbb{N}.$$

gebracht werden. Die Restriktionen werden von vielen Optimierungsverfahren benötigt.

3.1 Funktionen einer Veränderlichen

3.1.1 Differenzierbare Funktionen

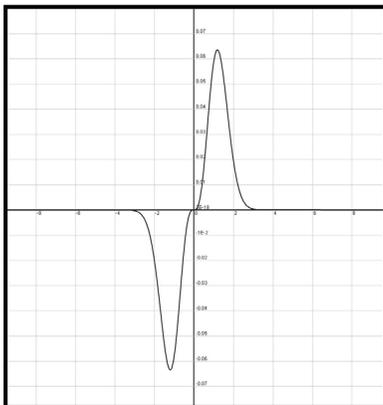


Abbildung 3.1: Brent's fünftes Problem

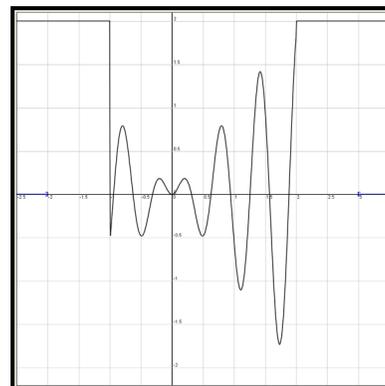


Abbildung 3.2: Michalewicz's erstes Problem