

Bachelorarbeit

Performance-Evaluation von PSO anhand der Maßsynthese eines Roboters

Oskar Pusz

Hannover, den 4. März 2014

LEIBNIZ UNIVERSITÄT HANNOVER
FAKULTÄT FÜR ELEKTROTECHNIK UND INFORMATIK
INSTITUT FÜR SYSTEMS ENGINEERING
FACHGEBIET SYSTEM- UND RECHNERARCHITEKTUR

Erstprüfer:	Prof. Dr.-Ing. Christian Müller-Schloer
Zweitprüfer:	Prof. Dr.-Ing. Bernardo Wagner
Betreuer:	Dipl.-Math. Sebastian Niemann

Gottfried Wilhelm Leibniz Universität Hannover,
Institut für Systems Engineering – System- und Rechnerarchitektur
Appelstr. 4, 30167 Hannover

Fakultät für Elektrotechnik
und Informatik

Bachelorarbeit

Fachgebiet System- und
Rechnerarchitektur (SRA)

Performance-Evaluation von PSO anhand der Maßsynthese eines Roboters

Sebastian Niemann

Tel. +49 511 762 19732
niemann@sra.uni-
hannover.de

Im Rahmen dieser Abschlussarbeit soll die Maßsynthese einer kinematischen Maschine mithilfe populationsbasierter Verfahren untersucht werden.

Stand: 30.04.2013

Das Ziel der Arbeit umfasst dabei:

- Den Erkenntnisgewinn über Eigenschaften des vorgegebenen Optimierungsproblems zur Maßsynthese eines 3RUS Roboters.
- Die Untersuchung der generellen Eignung des Particle Swarm Optimisation (PSO) Algorithmus für dieses Problem.

Die Maßsynthese beschäftigt sich mit der Bestimmung von Abmessungen einer Maschine, welche ihr maximales Leistungspotenzial erreichen. Anwendung findet dies im Entwurf von leistungsfähigen, hochdynamischen Robotersystemen in Handhabungs- und Montageprozessen. Eine kinematische Maschine besitzt mehrere kinematische Ketten, welche aus einer Folge von miteinander verbundenen Gelenken bestehen.



Fertigungsstraße mit mehreren Parallelrobotern.

Zur Bearbeitung der Abschlussarbeit ist die Zielfunktion einer 3RUS Parallelkinematik geben. Die Abschlussarbeit kann sowohl in Java, als auch in C++ erfolgen.

Besucheradresse:
Appelstr. 4
30167 Hannover
www.sra.uni-hannover.de

Zentrale:
Tel.: +49 511 762 0

Erklärung

Hiermit versichere ich, dass ich die vorliegende Arbeit selbständig und ausschließlich mit Hilfe der angegebenen Quellen und Hilfsmittel angefertigt habe.

Hannover, den 4. März 2014

Oskar Pusz

Zusammenfassung

In dieser Bachelorarbeit wird herausgestellt, ob sich der parametrisierbare *Standard Particle Swarm Optimization 2011* Algorithmus für die Maßsynthese bezüglich der Stablängen des 3RUS Parallelroboters eignet. Es entsteht ein Optimierungsproblem, das anhand experimenteller Methoden auf bestimmte Eigenschaften näher untersucht wird. Anhand dieser ermittelten Eigenschaften wird eine Parameteranalyse durchgeführt um eine geeignete Parametrisierung für den *Standard Particle Swarm Optimization 2011* Algorithmus zu finden. Der *Standard Particle Swarm Optimization 2011* Algorithmus mit ermittelter Parametrisierung wird abschließend mit den parameterfreien Algorithmen *Zufallssuche*, *Hill Climber* und *Hooke-Algorithmus* verglichen um herauszustellen, wie effizient der *Standard Particle Swarm Optimization 2011* Algorithmus schlussendlich gegenüber simpleren Algorithmen ist und ob sich der Aufwand der Analyse überhaupt gelohnt hat.

Abstract

This bachelor thesis will expose whether the configurable *Standard Particle Swarm Optimization 2011* algorithm is eligible for the parameter synthesis regarding the link lengths of the 3RUS parallel robot. It comes into an optimization problem which will be examined with experimental methods for certain properties. Based on these determined properties a parameter analysis will be carried out to find an appropriate parameterization for the *Standard Particle Swarm Optimization 2011* algorithm. The *Standard Particle Swarm Optimization 2011* algorithm with determined parameterization is finally compared with the non-parametric algorithms *Random Search*, *hillclimber* and *Hooke-Algorithmus* to expose how efficient the *Standard Particle Swarm Optimization 2011* algorithm in contrast to simpler algorithms is and whether the effort of analysis has ever been worth it.

Inhaltsverzeichnis

1	Einführung	1
1.1	Motivation	1
1.2	Stand der Technik	1
1.3	Ziel der Arbeit	2
1.4	Struktur der Arbeit	3
2	Grundlagen	4
2.1	Parallelroboter	4
2.2	Maßsynthese	6
2.3	Optimierungsproblem	6
3	Beschreibung des Optimierungsproblems	8
3.1	Global Condition Index	8
3.2	Formulierung des Optimierungsproblems	11
3.3	Eigenschaften des Optimierungsproblems	11
4	Experimentelle Untersuchung der Zielfunktion	13
4.1	Separierbarkeit	14
4.1.1	Methodik	14
4.1.2	Durchführung	15
4.1.3	Ergebniss	17
4.2	Laufzeit der Zielfunktion	17
4.3	Uni- oder Multimodal	18
4.3.1	Methodik	18
4.3.2	Durchführung	19
4.3.3	Ergebniss	20
4.4	Steigungsverhalten	22
4.4.1	Methodik	22
4.4.2	Durchführung	23
4.4.3	Ergebniss	23
4.4.4	Untersuchung zur Suchraumgröße	25
4.5	Zusammenfassung	26

5	Evaluation des Standard Particle Swarm Optimization 2011	28
5.1	Systematik der Analyse	31
5.2	Ausgangssituation	32
5.3	Parameteranalyse	33
5.3.1	Faktoranalyse	33
5.3.2	Populationsgrößenanalyse	37
5.4	Vergleich	39
6	Schlussfolgerung	43
7	Ausblick	45
	Literaturverzeichnis	46
	Abbildungsverzeichnis	48
	Tabellenverzeichnis	49

1 Einführung

Übersicht

1.1	Motivation	1
1.2	Stand der Technik	1
1.3	Ziel der Arbeit	2
1.4	Struktur der Arbeit	3

1.1 Motivation

Seit Ende der 80er Jahre wurden Parallelroboter als Industrieroboter für den praktischen Einsatz untersucht und entwickelt. Die Entwicklung im Bereich der Parallelroboter versprach ein viel versprechendes Potenzial, sodass eine große Anzahl an Prototypen entstand. Die anfängliche Euphorie ließ rapide nach als klar wurde, dass die Konstruktion eines effizienten Parallelroboters mit den vorhandenen Methoden nicht zufriedenstellend geschehen kann, da entweder die Leistungssteigerung gegenüber seriellen Robotern niedriger oder der Aufwand der Konstruktion um einiges höher war als erwartet [12]. Methoden zur Konstruktionsplanung eines Parallelroboters sind unter Anderem die Struktursynthese, mit der geeignete Bauteile im Bezug der Aufgabenstellung analysiert werden, und die Maßsynthese, die Eigenschaften der jeweiligen Bauteile, wie Länge und Gewicht, bestimmt. Diese Methoden gestalten sich als sehr aufwendig, sodass folglich Optimierungsprobleme entstehen. In Anbetracht der gegenwärtigen Forschungsthemen, siehe Abschnitt 1.2, wird immer mehr nach effizienteren Methoden gesucht Parallelroboter in der Planung und Konstruktion weiter zu verbessern, da die Komplexität des Vorgehens, wie es in dieser Arbeit zu sehen wird, noch immer in einem hohen Maß existiert und reduziert werden muss um Parallelroboter für den industriellen Alltag zum Vorteil der jeweiligen Aufgaben und Kosten zu bauen.

1.2 Stand der Technik

Zu den meachanischen Kriterien eines Parallelroboters gehören unter Anderem der Positionsfehler des Endeffektors, um den es auch hier in der Arbeit geht, Größe des Arbeitsraums, Steifigkeit, Manipulierbarkeit des Parallelroboters, die es zu optimieren gilt [1][6][15]. Es wird ersichtlich, dass das Spektrum an Optimierungskriterien sehr groß ist und vereinzeld zu Optimierungsproblemen werden, die anhand effizienten Algorithmen gelöst werden sollten.

Es wurden unter Anderem klassische Abstiegsverfahren wie z.B. Gradienten- und Newton-Verfahren [4], der *Genetische Algorithmus* (GA) [1][2][3][12] oder der *Particle Swarm Optimization Algorithmus*[13][14] zur Optimierung genutzt.

Die Abschätzung des Positionsfehlers, um den es hier in dieser Arbeit gehen wird, wird allgemein anhand des gleichen Kriteriums optimiert.

In Kreffts Ausarbeitung [12] wurde der GA unter Anderem zur Optimierung des Positionsfehlers in Augenschein genommen. Es wurde auch der hier zu untersuchende Roboter analysiert und mit dem GA optimiert. Krefft zeigte, dass der GA hinsichtlich der qualitativen Lösungsfindung ein geeigneter Algorithmus sei.

In den Ausarbeitungen [13] und [14] wurde der PSO unter Anderem auch auf die Positionsfehleroptimierung untersucht. In diesen Ausarbeitungen wird mit einer Abwandlung des eigentlichen PSO gearbeitet. Die Abwandlung nennt sich *Multi-objective Particle Swarm Optimization* und unterscheidet sich im Hinblick auf die Aktualisierung der Informationen, der Verwaltung von Partikeln und die Wahl der führenden Partikel. Hier zeigte sich, dass der PSO im Allgemeinen bezüglich der Lösungsfindung und dem Konvergenzverhalten ebenfalls für die Optimierung von Parallelrobotern geeignet ist.

Beyer [4] erwähnt, dass die Wahl des Algorithmus stark von der jeweiligen betrachteten Funktion abhängig ist. Heuristische Verfahren, wie der GA oder PSO, zeigen bei einfacheren Funktionen zwar Konvergenzen zur Lösung, aber mit dem Nachteil der benötigten verhältnismäßig hohen Rechenzeit. Für einfachere Funktion bevorzugt Beyer Abstiegsverfahren. Somit liegt nahe, dass kein idealer Algorithmus für alle Optimierungsprobleme innerhalb der Optimierung von Parallelrobotern existiert.

In den o.g. Ausarbeitungen wird nicht explizit auf den Rechenaufwand der Algorithmen eingegangen und der in dieser Arbeit zu untersuchende Typ von Parallelrobotern ist nicht allgemein Fokus der Forschung, sodass hier Handlungsbedarf besteht. Ebenfalls ist der hier zu evaluierende *Standard Particle Swarm Optimization 2011* im Kontext der Optimierung von Parallelrobotern noch nicht untersucht worden, sodass die Evaluation des *Standard Particle Swarm Optimization 2011* einen Ausblick auf einen neuen Algorithmus für diese Art von Optimierungsproblemen gibt.

1.3 Ziel der Arbeit

Das Ziel dieser Arbeit ist die Analyse der Eignung des *Standard Particle Swarm Optimization 2011* für die Maßsynthese zur Genauigkeitssteigerung zu einem in dieser Arbeit behandelten Parallelroboters. Unter der Maßsynthese ist zu verstehen, dass die Maße bestimmter Bauteile

des Roboters weitestgehend auf bestimmte Eigenschaften hin optimiert werden sollen. In dieser Arbeit wird die Eigenschaft der Genauigkeit bzw. der durchschnittliche Positionsfehler des beweglichen Teils des Roboters optimiert. Es handelt sich dabei um ein Optimierungsproblem, dass anhand ausgewählter Methoden auf bestimmte Eigenschaften, wie z.B. das Steigungsverhalten oder die Anzahl an Optimalstellen, untersucht wird. Anhand der Eigenschaften des Optimierungsproblems soll dann der Standard Particle Swarm Optimisation 2011 Algorithmus feinjustiert werden, um im Vergleich mit anderen Algorithmen die generelle Eignung des Verfahrens für dieses Problem zu untersuchen.

1.4 Struktur der Arbeit

Als nächstes folgt Kapitel 2, das *Grundlagen* bezüglich der Thematik dieser Arbeit näher erläutert. Dies dient zum besseren Verständniss der folgenden Kapitel. Im Anschluss folgt Kapitel 3 mit einer *Beschreibung* des Optimierungsproblems und der dazugehörigen Zielfunktion. Die Zielfunktion wird anschließend im Kapitel 4 einer *Experimentellen Untersuchung* unterzogen um Indizien über wichtige Eigenschaften zu gewinnen. Ebenfalls folgt eine Ermittlung der benötigten Laufzeit zur Berechnung der Zielfunktion. Auf Basis der ermittelten Eigenschaften der Zielfunktion erfolgt im Kapitel 5 die *Evaluation des Standard Particle Swarm Optimization 2011 Algorithmus*. In diesem Kapitel wird untersucht, ob sich der Algorithmus für das Optimierungsproblem, das in dieser Arbeit behandelt wird, überhaupt eignet. Die Eignung des Algorithmus zeigt sich durch die Erzielung von qualitativ guten Ergebnissen in absehbarem Rechenaufwand. Um eine solche Aussage treffen zu können, folgt eine ausführliche Untersuchung über die möglichen Einstellungsmöglichkeiten des Algorithmus um die Güte der Lösung zu maximieren und die Anzahl an erforderlichen Funktionsevaluationen zu minimieren. Im Abschluss der Arbeit gibt es in eine *Schlussfolgerung* im Bezug des vorherigen Kapitels wie auch einen *Ausblick* auf mögliche weitere Untersuchungsaspekte.

2 Grundlagen

Übersicht

2.1 Parallelroboter	4
2.2 Maßsynthese	6
2.3 Optimierungsproblem	6

Dieses Kapitel dient als Orientierung zur Thematik dieser Arbeit. Zuerst wird auf Parallelroboter im Allgemeinen eingegangen und welche Merkmale diese Art von Roboter mit sich bringen. Anschließend wird erläutert was die Maßsynthese generell ist und was dies im Bezug auf Parallelroboter bedeutet. Abschließend wird der Parallelroboter 3RUS beschrieben und was die Maßsynthese anhand des 3RUS in dieser Arbeit zu bedeuten hat. Abschließend wird ausgeführt, was eine Fitness oder Fitnesslandschaft im Bezug auf Optimierungsprobleme bedeutet.

2.1 Parallelroboter

Parallelroboter allgemein haben als wesentliche Bestandteile einen Endeffektor, eine Basiseinheit und mindestens zwei kinematischen Ketten, die mit der Basiseinheit und dem Endeffektor verbunden sind. Die Basiseinheit wird zum Beispiel am Boden oder an der Decke befestigt. Bei seriellen Robotern sind alle Gelenke aktiv zu steuern. Die Besonderheit bei parallelen Robotern im Gegensatz zu seriellen ist, dass nicht alle Gelenke steuerbar sein müssen, da sich die restlichen Teile der kinematischen Ketten passiv mitbewegen, d.h. es existiert mind. 1 passives Gelenk, dass nicht gesteuert werden muss. [17]

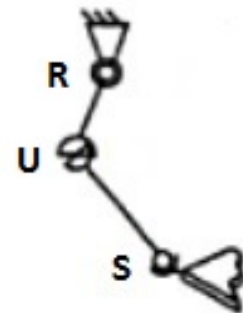


Abbildung 2.1: Kin. Kette RUS

In dieser Arbeit wird der Parallelroboter 3RUS untersucht. Die Basiseinheit wird an der Decke befestigt und 3 kinematische Ketten hängen herunter, die am anderen Ende am Endeffektor befestigt sind. Jede kinematische Kette besteht von oben nach unten aus einem Rotations- (**R**), einem Kreuz- (**U**niversal joint) und Kugelgelenk (**S**pherical joint). Die Unterstreichung des Buchstaben **R** sagt aus, dass es sich hier um ein aktives Rotationsgelenk also einen Rotationsmotor handelt und die nicht unterstrichenen Buchstaben stehen für passive Gelenke. Eine visuelle Veranschaulichung einer kinematischen Kette kann der Abbildung 2.1, die aus der Dissertation von Krefft [12] stammt, entnommen werden. Der Untersuchungsaspekt dieser Arbeit beinhaltet die

Optimierung aller Stablängen im Bezug auf den durchschnittlichen Positionsfehler der Endeffektorposition über den gesamten Arbeitsraum. Zur Veranschaulichung dient die Skizzendarstellung 2.2 eines 3RUS, basierend auf [11], in der die zu optimierenden Stablängen rot markiert sind.

Die Endeffektorposition wird allgemein als Pose bezeichnet und ist als 6 dimensionale Koordinate dargestellt:

$$\text{Pose} = (x, y, z, \alpha, \beta, \gamma)$$

Hierbei geben x, y, z die kartesische Position und α, β, γ die Roll-Nick-Gier-Winkel des Endeffektors, ausgehen von dessen Mittelpunkt, an. Im Folgenden werden nur die kartesischen Positionen variiert und die Roll-Nick-Gier-Winkel werden als konstant mit 0° Neigung betrachtet.

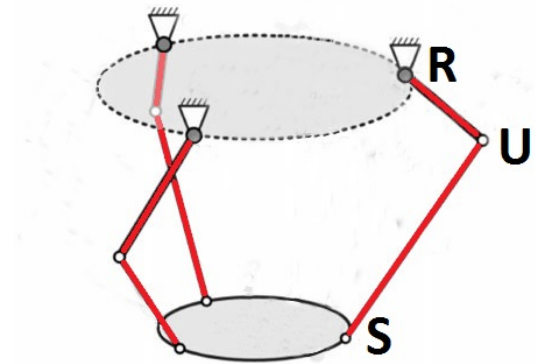


Abbildung 2.2: Skizze des 3RUS mit markierten Stäben



Abbildung 2.3: Flexpicker von ABB Automatics als annähernden Vergleich zum 3RUS

Als letzten Aspekt ist noch der Arbeitsraum, also der Raum aller möglichen Endeffektorpositionen, zu erläutern. Jedes Ende einer kinematischen Kette des 3RUS, welches am Endeffektor befestigt wird, spannt von der Basiseinheit aus einen Raum an möglichen Positionen auf. Der Raum hat aufgrund der 2 dimensional Rotation des Rotationsgelenks und der 3 dimensional Rotation des Kreuzgelenks die Form eines Torus. Bei den drei kin. Ketten entstehen also drei Tori aus möglichen Positionen der jeweiligen Enden jeder kin. Kette. Die Schnittmenge der 3 Tori ist folglich der gesamte Arbeitsraum des 3RUS. Somit existieren valide Posen nur innerhalb dieser Schnittmenge und alle möglichen Posen außerhalb dieser Schnittmenge sind nicht gültig, da sie niemals vom Roboter angefahren

werden könnten.

Ein Parallelroboter, der in der Industrie präsent ist und dem 3RUS am nächsten kommt, ist der sogenannte *Flexpicker* von *ABB Robotics*. Der *Flexpicker* hat im Unterschied zum 3RUS zwei Stäbe in dem unteren Teil der kin. Ketten und einen (vierten) Mittelstab. Durch die Mittelstange kann der Endeffektor des *Flexpickers* nicht in sich gedreht werden, wodurch der Endeffektor immer gerade herunter schaut, was beim 3RUS, durch die fehlende Mittelstange, nicht der Fall ist und Rotationen ansich möglich sind. Bis auf die beiden strukturellen Unterschiede und der Einschränkung der Rotation des Endeffektors beim *Flexpicker* sind die beiden Arten Roboter aber sonst vom Aufbau her vergleichbar.

2.2 Maßsynthese

Die Maßsynthese handelt von der Optimierung der Maße gegebener Strukturen bzgl. eines geforderten Kriteriums. Strukturen sind hier die einzelnen Komponenten des Roboters, wie Stäbe und der Endeffektor. Folglich können Strukturmaße zum Beispiel die Längen der Stäbe in kinematischen Ketten oder die Größe des Endeffektors sein. Kriterien können unter anderem folgende sein:

- Volumen des Arbeitsraums
- Positionsfehler des Endeffektors
- Kosten (Produktionskosten, Energiebedarf etc.)

Es handelt sich bei der Maßsynthese um eine globale Optimierungskomponente, die zu jeder Zeit und in jedem Einsatzfeld gelten soll. Die Globalität hat zur Folge, dass es sich hierbei um ein Optimierungsproblem handelt, was mit effizienten Methoden zur Lösung des Optimierungsproblems annähern soll. [17]

2.3 Optimierungsproblem

Bei einer kontinuierlichen, reellwertigen Optimierung existiert ein Suchraum $\Sigma \in \mathbb{R}^n$ und ein Lösungsraum $\Lambda \in \mathbb{R}$. Eine Zielfunktion f ist die Abbildung der Form $f : \Sigma \rightarrow \Lambda$. Bei Optimierungen wird, je nach Aufgabenstellung, innerhalb des Suchraums der Zielfunktion das Minimum oder Maximum des Lösungsraums gesucht, sodass es sich in der mathematischen Definition hier um ein Suchproblem handelt. Die Suche kann sich unter anderem bei hoher Dimension des Suchraums als sehr aufwändig gestalten, sodass effiziente Algorithmen angewendet werden

müssen um in absehbarer Zeit eine geeignete Lösung in Λ zu finden. Ein globales Optimum zu finden ist bei vielen Optimierungsproblemen eine utopische Vorstellung, da sie meist eine Zielfunktion behandeln, die schwer handelbare Eigenschaften wie starke Steigungen oder eine hohe Anzahl an lokalen Optimalstellen aufweisen, was die Suche nach einem globalen Optimum als sehr aufwendig bis unmöglich gestaltet.

Um Optimierungsprobleme händelbarer zu machen, wird in vielen Fällen ein Optimierungsproblem als Fitnesslandschaft abgebildet. Eine Fitness ist das Bild (also die Projektion) der Lösung aus Λ , das i.d.R. im Intervall $[0, 1] \in \mathbb{R}$ liegt. Dies hat den Vorteil, dass die möglichen Lösungen, die gefunden werden, vom Wertebereich her überschaubar bleiben und die Qualität einer Lösung besser eingeschätzt werden kann. Die Fitnesslandschaft ansich bildet sich also aus dem bekannten Suchraum Σ und der dazugehörigen Projektion des Lösungsraums in Form der Fitness F ab:

$$\textit{Fitnesslandschaft} : \Sigma \rightarrow F$$

Fitnesslandschaften können in der Hinsicht, des überschaubaren Lösungsraums besser analysiert werden. Für die Analyse von Fitnesslandschaften existieren eine Vielzahl an Methoden, die Eigenschaften der Fitnesslandschaften bestimmen können. Eigenschaften sind unter anderem Stetigkeit, Differenzierbarkeit, Steigungen, Anzahl an (lokalen) Optima wie auch der generelle Wertebereich der Fitnesslandschaft. Anhand dieser Eigenschaften können Algorithmen in Hinsicht ihrer Effizienz zur Lösungsfindung ggf. eingeschätzt oder auch modifiziert werden.

3 Beschreibung des Optimierungsproblems

Übersicht

3.1	Global Condition Index	8
3.2	Formulierung des Optimierungsproblems	11
3.3	Eigenschaften des Optimierungsproblems	11

Im folgendem Kapitel, welches als Grundlage zum Verständnis der Untersuchung dient, wird das Optimierungsproblem dieser Arbeit dargestellt. Hauptaugenmerk des Optimierungsproblems und der weiteren Betrachtung der Arbeit ist der sogenannte Global Contition Index, der im Folgenden genauer erläutert wird. Als Abschluss dieses Kapitels werden einige analytisch betrachtete Fakten dieser Funktion beschrieben.

3.1 Global Condition Index

Für den Anfang der Betrachtung wird die kinematische Jacobimatrix J des 3RUS miteinbezogen. Sie ist abhängig von den betrachteten Stablängen und der zu betrachteten Endeffektorpose des 3RUS. Die Jacobimatrix hat im Allgemeinen folgende Gestalt:

$$J = \begin{bmatrix} \frac{\partial G_1}{\partial x} & \dots & \frac{\partial G_1}{\partial \gamma} \\ \vdots & \ddots & \vdots \\ \frac{\partial G_i}{\partial x} & \dots & \frac{\partial G_i}{\partial \gamma} \end{bmatrix} \quad (3.1)$$

Die kin. Jacobimatrix J beschreibt das Übertragungsverhalten der Parallelstruktur und wie sich infinitesimale Änderungen der Antriebspositionen auf die Endeffektorpose auswirken werden. Der Index i steht für die Anzahl der Übertragungsfunktionen in Abhängigkeit des Freiheitsgrads. Der Freiheitsgrad des 3RUS entspricht dem maximalen Freiheitsgrad von 6, sodass es sich hier um eine 6×6 - Matrix handelt.[12]

Als nächstes wird auf die Konditionszahl dieser Matrix eingegangen. Die Konditionszahl κ einer Matrix bestimmt im Allgemeinen den Fehlerfortpflanzungsfaktor für weitere Berechnungen. Dieser Wert liegt innerhalb des Intervalls $[1, \infty)$. Die Konditionszahl der kin. Jacobimatrix lässt sich bei Parrallelstrukturen auf ihre Präzision des Endeffektors mit übergebenen Stablängen und einer gegebenen Pose übertragen. Je kleiner dieser Wert ist, desto stabiler steht der Endeffektor an dieser Pose. Um die Unendlichkeit des Intervalls zu umgehen wird der Kehrwert genutzt, sodass der Rechenwert innerhalb des Intervalls $(0, 1]$ liegt.

Die Konditionszahl ist hierbei eine Approximation des Positionsfehler des Endeffektors einer Parallelstruktur an einer vorgegebenen Pose [15]. Um den Positionsfehler über den gesamten Arbeitsraum zu bewerten, wurde der *Global Condition Index* (GCI) definiert. Der GCI entspricht dem Durchschnittswert der Konditionszahlen über den durch eine Stablängenkonfiguration aus der Menge aller Stablängenkonfigurationen L entstandenen Arbeitsraum der Parallelstruktur. Folglich lautet die Formel für den GCI für die Anzahl gemessender Posen im Arbeitsraum n [15]:

$$\text{GCI} = \frac{\sum_{i=1}^n \frac{1}{\kappa(J_{(l_{11}, \dots, l_{ij})}(\text{pose}_i)})}{n} \quad (3.2)$$

mit $J_{(l_{11}, \dots, l_{ij})}(\text{pose}_i)$ als die Jacobimatrix zu einer Stablängenkonfiguration $(l_{11}, \dots, l_{ij})$ und einer einzelnen Pose pose_i der insgesamt n ermittelten Posen. Für jede der einzelnen Bestimmung der Konditionszahlen ist eine neue Jacobimatrix zu bilden, da immer eine neue Pose in Bezug genommen wird. Es ist von Vorteil den entstandenen Arbeitsraum dahingehend zu untersuchen, dass der gesamte Arbeitsraum in gleichmäßig verteilten Posen ausgewertet wird. Allgemeine Zufallsposen bieten sich hier nicht an, da die Gefahr besteht, dass es zu einer Clusterung der betrachteten Posen kommt. Stattdessen sind Quasi-Zufallszahlen eine gute Betrachtung, da die dadurch entstehenden Posen den Untersuchungsraum gleichmäßig in einer Gitterstruktur unterteilen. In dieser Arbeit wird die sogenannte *van der Corput-Sequenz* (VDC-Sequenz) genutzt um Quasi-Zufallszahlen zu erzeugen. Die allgemeine VDC-Sequenz zur Basis 2 enthält Zahlen der Form:

$$1, \frac{1}{2}, \frac{1}{4}, \frac{3}{4}, \frac{1}{8}, \frac{5}{8}, \frac{3}{8}, \frac{7}{8}, \frac{1}{16}, \dots \quad (3.3)$$

Der erste Wert wird in dieser Arbeit nicht genutzt, da sie praktisch keinen Nutzen zum Aufbauen einer Gitterstruktur mit der VDC-Sequenz hat, sodass wenn von der VDC-Sequenz gesprochen wird, immer die Sequenz ohne die 1 gemeint ist. In Abbildung 3.1 kann der Unterschied zwischen einer beispielhaften uniformen und einer mit der VDC-Sequenz erzeugten quasi-zufälligen Verteilung gesehen werden. Jede Dimension der Posen in der Betrachtung des GCI kann anhand dieser Sequenz folglich gleichmäßig verteilt betrachtet werden, sodass die Rechnung mit einer gleichmäßigen Verteilung der Punkte innerhalb des Arbeitsraums durchgeführt werden kann. Die Größe der Sequenz in Abhängigkeit der Anzahl von gleichmäßigen Dimensionsaufspaltungen n beträgt

$$\sum_{i=0}^n 2^i, \quad n \geq 0$$

Elemente, was gleichzeitig der Anzahl in die Berechnung des GCI miteinbezogenen Pose-Stellen in jeder Dimension entspricht.

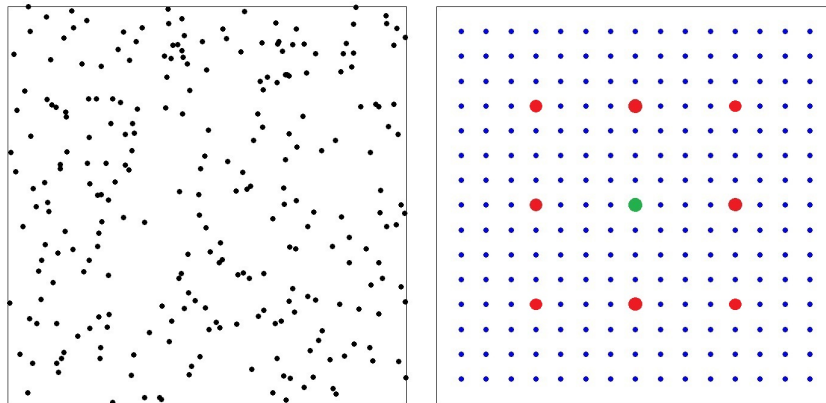


Abbildung 3.1: Veranschaulichung einer uniformen und quasi-zufälligen Verteilung

Es folgt mit Hilfe der Abbildung 3.1 eine etwas beispielhaftere Erklärung wie die quasi-zufällige Verteilung entsteht und was hier mit gleichmäßiger Dimensionsaufspaltung gemeint ist:

- Für $n = 0$ existiert nur $\langle \frac{1}{2} \rangle$ in der VDC-Sequenz. Es wird keine Dimension aufgespalten und folglich der generelle, einzige Mittelpunkt des gesamten Suchraums ausgewertet. (Grüner Punkt)
- Für $n = 1$ existieren $\langle \frac{1}{2}, \frac{1}{4}, \frac{3}{4} \rangle$ in der VDC-Sequenz. Nun wird nach der Auswertung des generellen Mittelpunkts ($\frac{1}{2}$) die Dimension mittig aufgespalten und in diesen Teilintervallen die jeweiligen Mittelpunkte ($\frac{1}{4}, \frac{3}{4}$) ausgewertet. (Rote Punkte)
- Diese Systematik geht für weitere n immer so weiter. Dimensionen werden aufgespalten und jeweils die Mittelpunkte aus den aufgespaltenen Dimensionen werden in die Berechnung einbezogen.

Im dreidimensionalen Raum, würde die o.g. Systematik für jede der drei Dimensionen einzeln durchgeführt werden um im gesamten Raum eine gleichmäßige Verteilung zu garantieren.

Durch diese Systematik wird vermieden, dass Bereiche mit ungültigen Posen, also außerhalb des Arbeitsraums, im Vergleich zur uniformen Verteilung mehrfach abgefragt werden. Eine Clusterung außerhalb des Arbeitsraums der uniformen Verteilung hätte zur Folge, dass sich der Rechenaufwand unnötig erhöhen würde, da ungültige Posen praktisch keinen Nutzen haben, aber dann vermehrt verwaltet werden müssen. Um dieses Risiko zu umgehen wird die *van der Corput-Sequenz* genutzt. Mit der quasi-zufälligen Verteilung wird jeder Bereich innerhalb des Arbeitsraum zuverlässig nur ein Mal abgefragt.

3.2 Formulierung des Optimierungsproblems

Wie in Abschnitt 2.2 beschrieben, handelt es sich bei dieser Arbeit um die Betrachtung der Maßsynthese im Bezug auf den GCI als Wert für die Güte der Stablängenkonfiguration des 3RUS. Der Wertebereich des GCI liegt wie aus dem vorherigen Abschnitt ableitbar im Intervall $(0, 1]$, wobei die 1 als höchster Wert der beste ist. Viele Ausarbeitungen im Bereich von Optimierungsproblemen behandeln Minimierungsprobleme. Um diese Ausarbeitung für andere Betrachtungsweisen von Optimierungsproblemen kompatibel zu machen, wird der Wert des entstandenen GCI negiert. Folglich existiert ein Wertebereich $[-1, 0)$, wobei die -1 als kleinster Wert nun der beste ist. Dies entspricht der Zielfunktion dieser Arbeit.

Das Optimierungsproblem der Maßsynthese am 3RUS liegt darin, dass aus der Menge aller möglichen Stablängenkonfigurationen L der beste Wert gefunden werden soll. Die konkrete Funktion lautet im Allgemeinen wie folgt:

$$\min_{l \in L} -\text{GCI}(l), \quad L = [l_{\min}, l_{\max}]^{i,j} \quad (3.4)$$

Der Index i steht für die Anzahl der Stäbe jeder kinematischen Kette und j für die Gesamtzahl an kinematischen Ketten der Parallelstruktur. Wie im Abschnitt 2.1 beschrieben besitzt der 3RUS 3 kinematische Ketten mit jeweils 2 Stäben. Somit handelt es sich dabei um ein 6 dimensionales Optimierungsproblem. Es werden für die weitere Untersuchung dieses Optimierungsproblems vorerst die Stablängenkonfigurationen $L \in [0.01, 1.00]^6$ in Metern betrachtet. Die minimale Länge von 0.01 Metern pro Stab ist ein äußerst minimalster Wert. Aber die Untersuchung von so kurzen Stablängen könnte ebenfalls zu interessanten Ergebnissen führen, sodass sich wichtige Erkenntnisse der Eigenschaften der Zielfunktion sammeln lassen könnten. Die maximale Stablänge von 1 Meter ergibt im ausgestreckten Zustand einer kinematischen Kette 2 Meter, was eine gute Reichweite des Roboters ergibt. Aber schon 1 Meter und erst recht höhere Werte bergen das Risiko der Instabilität des Endeffektors, sodass hier vorerst die maximale Länge von 1 Meter betrachtet wird.

3.3 Eigenschaften des Optimierungsproblems

Die Jacobimatrix (3.1) kann bei Parallelstrukturen oft nicht analytisch beschrieben werden, sodass auf andere Methoden ausgewichen werden muss. Auch im Bezug auf den 3RUS ist es nicht möglich, die Jacobimatrix analytisch zu bestimmen [12]. Auf die Beschreibung des Rechenweges, wird aufgrund der hohen Komplexität verzichtet, der in den Ausarbeitungen von Krefft [12] oder Kotlarski, Heimann und Ortmaier [11] nachgelesen werden kann. Zu erwähnen

ist aber, dass die Berechnung einer Konditionszahl über eine Singularwertzerlegung erfolgt. Die Singularwertzerlegung wird auf Matrizen angewandt und zerlegt die Matrix in ein Produkt aus drei Matrizen, was diese Operation nicht analytisch formulierbar macht.

Desweiteren handelt es sich bei (3.2) ebenfalls nur um eine numerische Annäherung, da aufgrund der Gitterstruktur der Arbeitsraum niemals vollständig ausgewertet wird. Eine Berechnung über Integrale des Arbeitsraums ist aufgrund der fehlenden analytischen Darstellung der Konditionszahl ebenfalls nicht möglich. Folglich ist die Zielfunktion des Optimierungsproblems nicht analytisch formulierbar.

Innerhalb des Rechenweges der Zielfunktion existieren viele trigonometrische Operationen. Diese Operationen müssen aufgrund der Berechnung der kinematischen Ketten und den Lagen und Neigungsmöglichkeiten ihrer Gelenke in der Rechnung enthalten sein. Da trigonometrischen Operationen global weder konvex noch konkav sind, ist die Zielfunktion des Optimierungsproblems ebenfalls nicht konvex oder konkav.

Das 6 dimensionale Optimierungsproblem ist also eine nicht-lineare, numerisch zu bestimmende Funktion. Diese Eigenschaften dieser Funktion bilden einen entscheidenden Nachteil: Es ist nicht möglich einen Gradienten zu bilden und demnach auf diesem Wege die Minimalstellen zu lokalisieren. Folglich wird im folgendem Kapitel auf ableitungsfreie Methoden zurückgegriffen um die Zielfunktion bzw. das Optimierungsproblem experimentell zu untersuchen.

4 Experimentelle Untersuchung der Zielfunktion

Übersicht

4.1	Separierbarkeit	14
4.2	Laufzeit der Zielfunktion	17
4.3	Uni- oder Multimodal	18
4.4	Steigungsverhalten	22
4.5	Zusammenfassung	26

In diesem Kapitel wird die Zielfunktion (3.2) und implizit das Optimierungsproblem experimentell anhand von vier grundlegenden Aspekten analysiert:

- Separierbarkeit
- Laufzeit
- Uni- / Multimodalität
- Steigungsverhalten

Es wird mit der Frage der Separierbarkeit begonnen, da es von Vorteil für weitere Betrachtungen wäre, wenn die 6 dimensionale Zielfunktion in Teildimensionsräume aufgespalten werden könnten. Dies würde in allen Betrachtungspunkten dieser Arbeit bzw. auch in zukünftigen Untersuchungen zum 3RUS zu einer deutlich geringerer Komplexität der Funktion und folglich zu geringerem Rechenaufwand führen.

Anschließend wird auf den Rechenaufwand eingegangen um nachfolgende Untersuchungen besser gewichten zu können um zu hohe Laufzeiten zu vermeiden. Es werden verschiedene Laufzeitmessungen zur Berechnung der Zielfunktion in Java- wie auch gcc-Umgebung dargestellt.

Um zu untersuchen, ob die betrachtete Zielfunktion uni- oder multimodal ist, wird die *Fitness Distance Correlation* [10] genutzt, die in einem folgenden Abschnitt weiter erläutert wird. Modalität kann als die Anzahl von Optimalstellen innerhalb der Funktion verstanden werden. Ist eine Funktion unimodal existiert ein einziges globales Optimum und ist eine Funktion multimodal existieren zusätzlich davon noch lokale Optima.

Als letztes wird unterstützend dazu ebenfalls auf die Methodik der *Fitness Clouds* [16] zurückgegriffen, die Rückschlüsse auf Nachbarpunkte eines betrachteten Punktes geben wird. Mit Hilfe

der *Fitness Clouds* können Schlüsse auf mögliche Steigungen innerhalb der Fitnesslandschaft gezogen werden.

4.1 Separierbarkeit

In diesem Abschnitt wird die Methodik, Durchführung und das Ergebniss der Überprüfung zur Separierbarkeit der Zielfunktion und folglich des Optimierungsproblems dargestellt. Die Separierbarkeit ist eine Eigenschaft, die die Komplexität einer Funktion reduziert. Sie kann besagen, dass z.B. die 6 dimensionale Zielfunktion dieser Arbeit in drei 2 dimensionale Teilräume separierbar ist, ohne dass sich die Funktionswerte ändern. Diese Eigenschaft würde den Rechenaufwand um ein vielfaches verringern, da nur noch innerhalb der jeweiligen Teilräume evaluiert werden muss und dadurch die Komplexität der Funktion deutlich reduziert wird.

4.1.1 Methodik

Um die Separierbarkeit zu prüfen wird die folgende Gleichung betrachtet und auf ihre Allgemeingültigkeit geprüft:

$$f(\arg \min_{X \in \mathbb{R}^6} f(X)) = f(\delta[\arg \min_{Y_1 \in \mathbb{R}^n \times \text{const}^{6-n}} f(Y_1), \arg \min_{Y_2 \in \mathbb{R}^{6-n} \times \text{const}^n} f(Y_2)]) \pm \epsilon \quad (4.1)$$

mit n für die Anzahl variabler Dimensionen und const für einen beliebigen konstanten Wert. In anderen Worten simpel ausgedrückt werden die Argumente eines Minimums im vollen 6 dimensional Suchraum mit allen dimensional komplement zueinander stehenden Teilräume des 6 dimensional Raums auf Gleichheit ihres Funktionswertes geprüft. δ steht für sämtliche Permutationen dieser komplementären Teilräume. Sind alle Permutationen ungültig, ist die Zielfunktion nicht separierbar. Ist die Funktion in permutierte Teilräume separierbar, muss auf Teilseparationen geprüft werden. Um die rechte Seite der Gleichung bei jeder Permutation zu erzeugen, werden die jeweils komplementär zueinander nicht zu beachtenden Dimensionen in beiden Teilen auf einen konstanten Wert gesetzt. So existieren variable und konstante Dimensionen mit denen nach einem Minimum der Zielfunktion gesucht wird. Sind die Argumente auf beiden Seiten der Gleichung gefunden, werden die Funktionswerte miteinander verglichen. Sind diese Funktionswerte zu einer bestimmten Toleranz ϵ gleich, ist diese Funktion in die Teilräume bzgl. dem Verhältnis der variablen und konstanten Dimensionen separierbar. Das ϵ dient zur Vermeidung des Einflusses von möglichen Rundungsfehlern bei der Berechnung durch den Rechner.

4.1.2 Durchführung

Unterstützend zur Beschreibung der Durchführung dient der folgende vollständige Permutationsbaum als visuelle Veranschaulichung. Auf die Färbung der einzelnen Knoten wird im Verlauf des Abschnitts eingegangen.

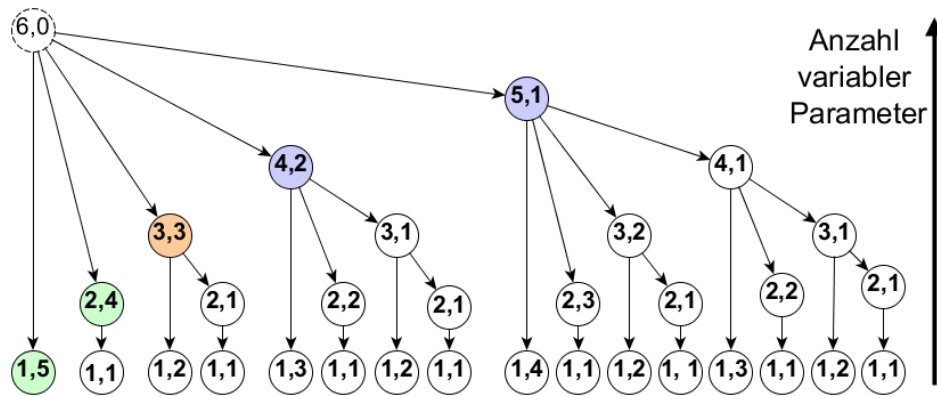


Abbildung 4.1: Vollständiger Permutationsbaum zur Separierbarkeitsüberprüfung

Es folgt die Erklärung des Grundprinzips der Durchführung. Jeder Knoten symbolisiert einen 2-Tupel der Form (x, y) . x steht für die Anzahl variabler Dimensionen und y für die Anzahl konstant gehaltenen Dimensionen. Dieses Verhältniss der variablen zu den konstanten Dimensionen soll sich auf den ersten Teil der Permutationsbetrachtung in der rechten Seite der Gleichung (4.1) beziehen. Die Wurzel des Baumes ist die Suche nach dem Minium im gesamten 6-dimensionalen Suchraum, was auf der linken Seite von (4.1) zu sehen ist. Anschließend soll für jeden Kindknoten der Wurzel der erste Teil der rechten Seite der Gleichung konstruiert, das Komplement im zweiten Teil gebildet und die Gleichung auf Gültigkeit geprüft werden. Ist eine Gleichung nach Konstruktion der Permutation gültig, werden die folgenden Kindknoten getrachtet um den gültigen Teilraum auf weitere Separationen zu untersuchen. Ist die Gleichung zu einer Permutation ungültig, kann sofort eine Permutation mit den Informationen des Nachbarknoten gebildet werden. Die folgenden Kindknoten müssen nicht mehr betrachtet werden, da die Funktion dann nicht weiter in kleinere Dimensionen aufspaltbar wäre. Ist kein Nachbar mehr vorhanden, wird der nächste Kindknoten der Wurzel betrachtet bis alle Permutationen durchlaufen sind.

Bei dieser simplen Systematik wären 24192 Permutationen zu bilden und einzeln auf ihre Gültigkeit zu prüfen. Dies lässt sich um ein hohes Maß reduzieren.

Wenn die Kindknoten der Wurzel von links nach recht abgearbeitet werden, sind die Betrachtungen der Teilräume, das entspricht den weißen Knoten in Abbildung 4.1, unnötig. Wenn zum Beispiel eine gültige Separation in zwei Teilräume der Größe 3 gefunden wurde (also war Knoten (3,3) gültig), dann würde simpel gedacht die 3 folglich als (1,2) und (2,1) geprüft werden, was von der Gesamtzahl an variablen Dimensionen in vorherigen Nachbarknoten aber schon geschehen ist. Somit werden alle weißen Knoten im Permutationsbaum unnötig. Ebenfalls ist die Beachtung des komplementären Verhaltens der Separierbarkeitfunktion (4.1) von Wichtigkeit, wodurch die blauen Knoten unnötig werden, da in anhand der grünen Knoten (1,5) und (2,4) automatisch (5,1), (4,2) mitgeprüft werden. Somit reduziert sich der Permutationsaufwand auf insgesamt $\binom{6}{1} + \binom{6}{2} + \binom{6}{3} = 41$ Permutationen, was einer Reduzierung von ca. 99,83% entspricht.

Der reduzierte Permutationsbaum halt also folgende Gestalt:

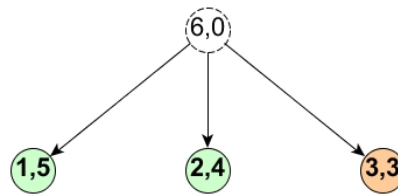


Abbildung 4.2: Reduzierter Permutationsbaum zur Separierbarkeitsüberprüfung

Anhand der oben ausgeführten Systematik können sämtliche mögliche Separationen überprüft werden. Wenn die Zielfunktion, um das oben erwähnte Beispiel aufzugreifen, in die Teilräume $T_1 = \{\{x_1, x_2\}, \{x_3, x_4, x_5, x_6\}\}$ und $T_2 = \{\{x_1, x_2, x_3, x_4\}, \{x_5, x_6\}\}$ separierbar ist, ist implizit eine Separation in $T = \{\{x_1, x_2\}\{x_3, x_4\}\{x_5, x_6\}\}$ möglich.

Zu Anfang wird der gesamte Suchraum $[0.01m, 1.00m]^6$ diskret nach eine besten Lösung durchsucht. Dann folgt der nächste Schritt: Die variablen Dimensionen werden im Wertebereich $[0.01m, 1.00m]$ diskret durchlaufen und die anderen Dimensionen bleiben bei einem konstanten Wert. Es werden die jeweiligen Argumente gesucht, die das Minimum der Lösung in diesem Kontext bilden. Diese Argumente werden gespeichert und die Prüfung wird mit dem Komplement fortgesetzt, d.h. variable Dimensionen werden konstant und konstante Dimensionen werden variabel. Dadurch entstehen 6 Argumente die in ihren jeweiligen Teilräumen die beste Lösung abbilden. Die Lösung aus dem vollen 6 dimensional Suchraum wird mit der Lösung mit den Argumenten aus den Teilsuchräumen verglichen. Sind diese Lösungen im Rahmen von ϵ gleich,

ist die Zielfunktion in die jeweiligen Teilräume separierbar. Es folgt kontinuierlich die Prüfung aller Permutationen bis alle verarbeitet wurden.

4.1.3 Ergebniss

Bei der Durchführung des Tests zur Separation wurde der konstante Wert jeweils mit 0.5 Meter definiert und ein ϵ von 0.001 akzeptiert.

Es wurde anhand der o.g. Methode ermittelt, dass die Funktion in allen Permutationen nicht separierbar ist. Somit bleibt die Zielfunktion eine 6 dimensionale Funktion, die sich in keiner Hinsicht in Teilräume aufspalten lässt und behält weiterhin ihre Komplexität.

4.2 Laufzeit der Zielfunktion

Die Laufzeit der Zielfunktion ist abhängig von der gewählten Größe der *van der Corput-Sequenz* (3.3), die den Raum der möglichen Posen zu einer Stäblängenkonfiguration in Gitterstruktur überprüft und von der Dimensionsgröße der Jacobimatrix, da jede der enthaltene Gleichungen für die Zielfunktion ausgewertet werden müssen. Die Dimension der Jacobimatrix ist in der hier getrachteten Zielfunktion konstant 6×6 , sodass nur die Gitterstruktur variiert werden kann.

Die Größe der Sequenz für eine gleichmäßige Aufteilung in Gitterstruktur des 3 dimensional Raums ist $(\sum_{i=0}^n 2^i)^3$, $n \geq 0$. Es werden Laufzeitmessergebnisse exemplarisch für die *van der Corput-Sequenzen* der Größen $\sum_{i=0}^n 2^i$, $n = \{0, 1, \dots, 6\}$ dargestellt. Bei der 3 dimensional Zielfunktion liegt das Spektrum von gemessenen Punkten also bei $[1^3, 127^3]$ Punkten.

Es folgt die Übersicht in Tabelle 4.1 der gemessenen Zeiten anhand zwei unterschiedlicher Systeme. Es wurde eine Durchschnittslaufzeit über 10000 Messungen für die exemplarischen Stablängen (0.35, 0.35, 0.35, 0.8, 0.8, 0.8) betrachtet. Zur Veranschaulichung der Gitterstruktur sind jeweils die kartesischen Distanzen zwischen den einzelnen Messpunkten pro Dimension und die Anzahl der Messpunkte im 3 dimensional Raum angegeben.

Wie zu sehen ist, steigt die Laufzeit mit steigendem n rapide an. Ab $n = 2$ ist der Faktor mit der sich die Laufzeit bei steigendem n erhöht sehr groß, sodass im Bezug der Laufzeit abgewogen werden muss wie hoch n für die Bestimmung des GCI gewählt werden sollte. Generell ist zu sagen, dass hier aufgrund des hohen Rechenaufwands der Fokus auf einen überschaubaren Rechenaufwand liegen wird. In den folgenden Untersuchungen wird man sehen, wie oft die Bestimmung des GCIs in den jeweiligen Methoden erfolgen muss, sodass eine zu hohe Laufzeit der Berechnung der Zielfunktion schnell in unermessliche Laufzeiten der jeweiligen Methoden geraten kann.

Prozessor	n	Java	gcc	Distanz	# Messpunkte
Intel Core i5-3570K	0	0.046 ms	0.013 ms	NIL	1 ³
	1	0.059 ms	0.033 ms	1/4	3 ³
	2	0.167 ms	0.096 ms	1/8	7 ³
	3	1.100 ms	0.901 ms	1/16	15 ³
	4	9.220 ms	6.975 ms	1/32	31 ³
	5	67.520 ms	53.868 ms	1/64	63 ³
	6	518.216 ms	421.934 ms	1/128	127 ³
Cluster (1 Prozessor)	0	0.085 ms	0.021 ms	NIL	1 ³
	1	0.121 ms	0.071 ms	1/4	3 ³
	2	0.323 ms	0.191 ms	1/8	7 ³
	3	2.162 ms	1.769 ms	1/16	15 ³
	4	16.320 ms	12.462 ms	1/32	31 ³
	5	127.698 ms	103.926 ms	1/64	63 ³
	6	1011.795 ms	819.395 ms	1/128	127 ³

Tabelle 4.1: Laufzeiten der Zielfunktion

Im weiteren Verlauf der Arbeit wird für die Bestimmung des GCIs $n = 4$ gewählt. Für $n \geq 5$ wäre zwar die Auflösung des Gitters steigend besser, aber die Rechenzeit in der Summe viel zu hoch. Für $n = 3$ ist die Rechenzeit gering, aber bei nur 15 Stellen pro Dimension die Auflösung zu klein um qualitative Aussagen über den GCI machen zu können.

4.3 Uni- oder Multimodal

In diesem Abschnitt wird untersucht, ob die Zielfunktion uni- oder multimodal ist und wie sich die Korrelation verhält. Als Methode zur Bestimmung dieser Eigenschaft wird die Systematik der *Fitness Distance Correlation* (FDC) genutzt [10]. Es folgt die Erläuterung der Methodik, Durchführung und abschließend wird das Ergebniss der FDC dargestellt.

4.3.1 Methodik

Als ersten Schritt ist ein Optimum der zu untersuchenden Funktion zu bestimmen, welches als Ankerpunkt für die weiteren Schritte dienen soll. Als Nächstes folgt ein routinierter Ablauf bis eine bestimmte Anzahl an Schritten durchgeführt wurde. Für jeden Schritt werden von einem (zufällig) bestimmten Punkt die euklidische Distanz zum Optimum und die Fitness des Punkts bestimmt. Für eine Menge von Punkten werden im Laufe aller Schritte der FDC-Bestimmung

die dazugehörige Menge an euklidischen Distanzen $D = \{d_1, d_2, \dots, d_n\}$ und Fitnesswerten $F = \{f_1, f_2, \dots, f_n\}$ bestimmt. Jeder dieser Punkte lässt sich mit der Beziehung $FDC : D \rightarrow F$ zu einem Graphen plotten, sodass bestimmte Formen entstehen können und grafisch abgelesene Aussagen über die Korrelation der Funktion gemacht werden können.

Desweiteren kann über die o.g. Mengen D und F ein Korrelationskoeffizient r über folgende Formel berechnet werden:

$$r = \frac{c_{D,F}}{s_D \cdot s_F}$$

mit $c_{D,F}$ als Kovarianz von D und F , sowie den Standardabweichungen s_D und s_F zu den jeweils dazugehörigen Mengen. r liegt im Intervall $[-1, 1] \in \mathbb{R}$. Je mehr r zur 0 strebt, desto eher kann von einer multimodalen Funktion gesprochen werden, da keine Aussage über die Korrelation der Funktion gemacht werden kann. Ein r von 1 steht für eine maximale Linkskorrelation, d.h. es könnte z.B. folgende Aussage formuliert werden: *"Je geringer die Distanz zum Optimum, desto geringer die Fitness"*, was der idealen Aussage bezüglich einer Minimierungsfunktion entspricht. Ein r von -1 steht analog für die ideale Aussage einer Maximierungsfunktion.[10]

Das Optimierungsproblem dieser Arbeit (3.4) beschäftigt sich mit der Minimierung der Fitness im Bezug der möglichen Stablängenkonfigurationen. Folglich sollte im besten Fall ein r von 1 entstehen.

4.3.2 Durchführung

Um ein Optimum zu lokalisieren wird in dieser Arbeit ein modifizierter Hill-Climber Algorithmus verwendet. Die Modifikation besteht darin, dass der Suchraum der möglichen Stablängenkonfigurationen in gleichgroße Teilräume, im Folgenden Sektoren genannt, aufgeteilt und für sich untersucht wird. Am Anfang wird im vollen Suchraum der eigentliche Hill-Climber Algorithmus mit 1000 Schritten durchgeführt. Die beste gefundene Fitness dient als Referenzwert für die nächste Betrachtung. Der volle Suchraum wird nun in Sektoren aufgespalten. In jedem Sektor wird ebenfalls mit dem Hill-Climber Algorithmus mit 1000 Schritten nach einem Optimum gesucht und der Sektor mit der besten gefundenen Fitness lokalisiert. Es folgt ein Vergleich, ob die in diesem einen Sektor gefundene Fitness besser ist als der Referenzwert. Ist dies der Fall, wird die bessere Fitness als Referenzwert übernommen und der Sektor, wo dieser gefunden wurde, wird wieder in Teilräume aufgespalten und der Abgleich aller neuen Sektoren geht von vorne los. Der Algorithmus läuft bis keine bessere Fitness und die damit verbundene Stablängenkonfiguration gefunden werden konnte. Durch diesen Algorithmus wird der Suchraum in einer breiten Fläche in absehbarer Zeit abgesucht und ergab qualitativ gute Ergebnisse, sodass sich hierdurch

ein geeigneter Ankerpunkt für die FDC ergibt.

Mit diesem Optimum wird jetzt die *Fitness Distance Correlation* durchgeführt. Der Suchraum der möglichen Stablängenkonfigurationen wird, genau wie bei der Berechnung des GCI, mit Hilfe der *van der Corput-Sequenz* durchschritten und jede Stelle rechnerisch durch Bestimmung des FDC-Koeffizienten r verarbeitet. Die *van der Corput-Sequenz* bietet hier den Vorteil, dass die Fitnesslandschaft im Bezug auf den angenommenen Optimalpunkt gleichmäßig untersucht wird, was zur Analyse Zielfunktion mit der FDC zu einer allgemeineren Aussage bzgl. der Zielfunktion führt.

4.3.3 Ergebniss

Der modifizierte Hill-Climber Algorithmus wurde mehrere Male ausgeführt und konnte unter anderem die in der folgenden Tabelle angegebenen Optima bestimmen. Es sind gerundete Stablängen in Meter sowie die Fitness angegeben. Jede Spalte pro Messung der Stablängenkonfigurationen entspricht den Maßen einer kinematischen Kette und die jeweiligen Zeilen den Stäben innerhalb der kinematischen Kette von oben (an der Basiseinheit) nach unten (am Endeffektor) gesehen. Jedes der horizontalen Felder innerhalb der Tabelle steht für eine Ausführung des Hill-Climber Algorithmus.

$l_{11...21}$	$l_{12...22}$	$l_{13...23}$	Fitness
0.368	0.180	0.400	-0.788
0.026	0.385	0.394	
0.172	0.529	0.496	-0.661
0.150	0.503	0.497	
0.133	0.408	0.391	-0.479
0.131	0.381	0.391	
0.373	0.136	0.374	-0.454
0.126	0.387	0.384	
0.490	0.172	0.505	-0.454
0.036	0.395	0.505	

Tabelle 4.2: Exemplarische Ergebnisse des modifizierten Hill-Climber Algorithmus

Wie zu sehen ist, unterscheiden sich die möglichen Stablängenkonfigurationen in jeder Messung, wie auch deren Fitnesswerte. Der modifizierte Hill-Climber Algorithmus ist folglich natürlich ungeeignet zur Suche der globalen Optimalstelle. Da für die FDC aber nur ein gut gewählter

Ankerpunkt nötig ist um auf Korrelationen und Eigenschaften zu schließen, wird die erste Konfiguration aus der Tabelle mit der besten Fitness als Ankerpunkt für die weitere Untersuchung gewählt.

Für die weitere Untersuchung mit der FDC wird der Suchraum mit einer Van-Der-Corput-Sequenz der Größe $\sum_{i=0}^2 2^i$ gewählt, sodass $7^6 = 117649$ Stabkonfigurationen ausgewertet werden, was in einem überschaubaren Rechenaufwand bleibt. Aus der Abbildung 4.3 kann ge-

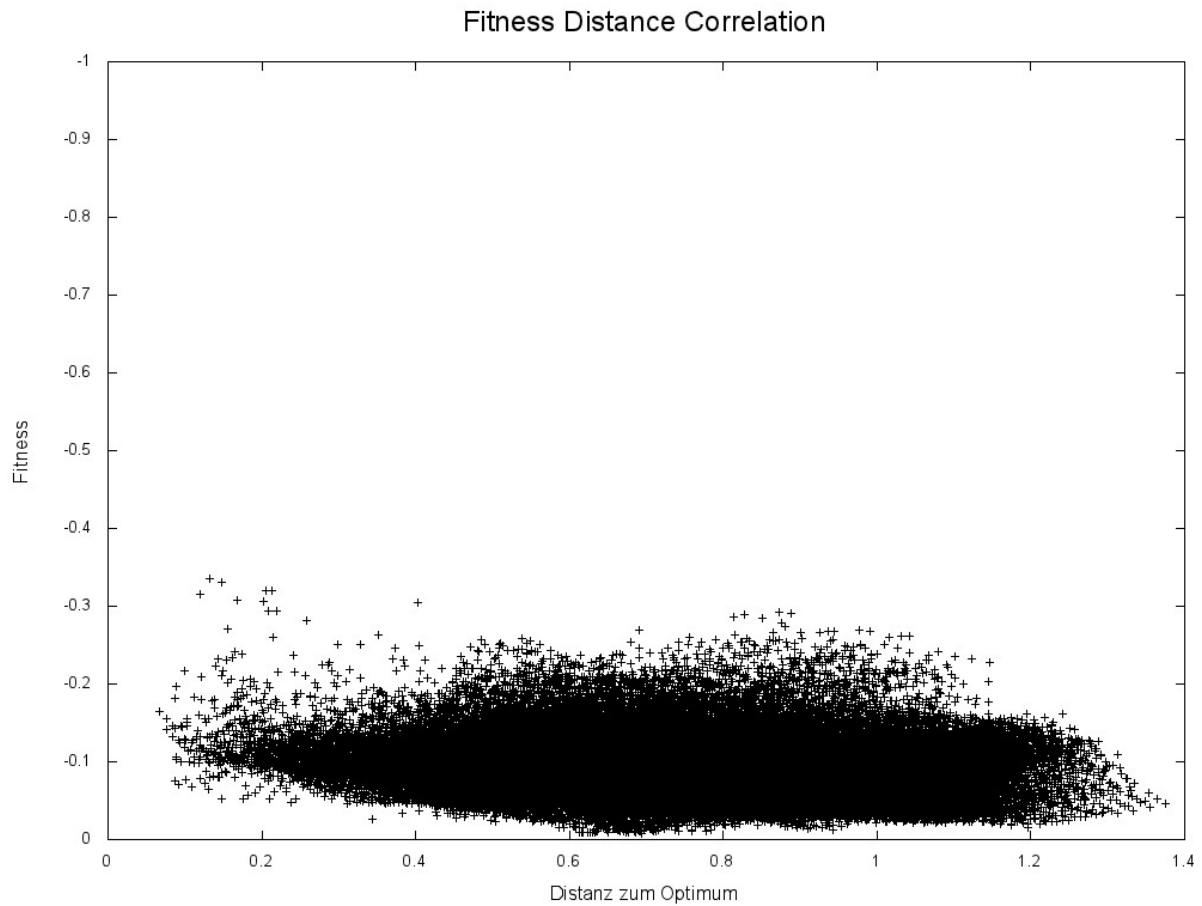


Abbildung 4.3: Plot der Fitness Distance Correlation

schlossen werden, dass dieses Optimierungsproblem definitiv multimodal sein muss. Aussagen wie in Abschnitt 4.3.1 können nicht formuliert werden. Die Darstellung lässt darauf schließen, dass die Zielfunktion nicht korreliert.

Der FDC-Koeffizient r für die oben dargestellte Grafik ist gerundet 0.041. Dieses r ist ein zweites Indiz dafür, dass die Zielfunktion nicht korreliert und multimodal sein muss, da r nicht in der Nähe der Randgebiete des Intervalls $[-1, 1]$ liegt. Für die Frage nach der Korrelation des

Optimierungsproblems ist die FDC ungeeignet. Zusätzlich ist zu sehen, dass die Fitnesswerte den Wert -0.4 im vollen Suchraum $[0.01m, 1.00m]^6$ nicht unterschreiten. Anhand der FDC lässt sich an dieser Stelle nicht deuten, weshalb die Zielfunktion diese Eigenschaft trägt.

4.4 Steigungsverhalten

In diesem Abschnitt wird mit Hilfe von *Fitness Clouds* (FC) die Fitnesslandschaft auf ihr Steigungsverhalten untersucht. Dadurch kann festgestellt werden, wie flach oder verrauscht die Fitnesslandschaft ist. Nach der Erläuterung der Methodik und Untersuchung folgt eine Ergebnisdarstellung, die anschließend näher auf verschiedene Teilsuchräume analysiert wird.

4.4.1 Methodik

Innerhalb der Fitnesslandschaft werden zufällig oder systematisch bestimmte Ausgangspunkte, im Folgenden Spots bezeichnet, betrachtet. Für jeden dieser betrachteten Spots $x_i \in X$ wird der Fitnesswert mit bekannter Funktion f und eine Menge von Nachbarpunkte wie auch deren Fitnesswerte mit Funktion $\tilde{f}$ bestimmt. $\tilde{f}$ beinhaltet die Bestimmung einer bestimmten Menge Nachbarpunkte wie auch die bekannte Funktion f zur Berechnung des Fitnesswertes für jeden einzelnen dieser Nachbarpunkte, sodass Tupel in der Form $(f(x_i), \tilde{f}(x_i))$ formuliert werden können. Alle diese Tupel lassen sich in der Beziehung $FC : f(X) \rightarrow \tilde{f}(X)$ zu einem Graphen abbilden.

Zusätzlich dazu wird üblicherweise für jeden Spot x_i aus der Menge an Nachbarfitnesswerten $\tilde{f}(x_i)$ ein Durchschnittssfitnesswert $\tilde{f}_{mean}(x_i)$ bestimmt. Mit allen $\tilde{f}_{mean}(x_i)$ lässt sich eine Durchschnittsfitnesslinie in der Beziehung $FC_{mean} : f(x_i) \rightarrow \tilde{f}_{mean}(x_i)$ innerhalb des Graphen abbilden. Mit dieser Durchschnittsline lässt sich abschätzen, wie das Durchschnittsverhalten in den Nachbarschaften aller gewählten Spots ist. Ebenfalls lässt sich dadurch sagen, wie wahrscheinlich es ist sich von einem bestimmten Spot aus zu verbessern oder zu verschlechtern, wenn Nachbarpunkte betrachtet werden sollen.

Als letzten Anhaltspunkt in der Betrachtung mit Hilfe von FC dient eine Ursprungsgerade. Mit dieser Ursprungsgerade kann untersucht werden, wie viele Nachbarschaftsfitnesswerte eines Spots über oder unter der Ursprungsgerade liegen. Damit lassen sich Aussagen treffen in welchen Spektrum die Fitness besser oder schlechter werden kann, wenn ein Nachbarschaftspunkt betrachtet werden soll. [16]

4.4.2 Durchführung

Die Funktion f ist die aus dem Abschnitt 3.2 bekannte Zielfunktion zur Bestimmung einer Fitness im Intervall $[-1, 0)$. Die Funktion $\tilde{f}$ bestimmt in dieser Betrachtung 100 Nachbarpunkte, die vom Spot aus innerhalb eines sphärischen Radius von 0.01m uniform verteilt bestimmt werden. Anschließend werden die Fitnesswerte aller bestimmten Nachbarn mit Funktion f ermittelt. Die Bestimmung der zu untersuchenden Spots erfolgt über uniform verteilte Zufallsvektoren, die eine Stablängenkonfiguration aus dem vollen Suchraum $[0.01\text{m}, 1.00\text{m}]^6$ abgebildet werden. Um genaue Aussagen über das Steigungsverhalten der Zielfunktion machen zu können, ist für die FC eine hohe Anzahl an Messpunkten erforderlich. Eine Verteilung über die Van-Der-Corput-Sequenz für die FC ist bezogen auf den Rechenaufwand in Abhängigkeit von n bei der Anzahl an Elementen in der Van-Der-Corput-Sequenz $\sum_{i=0}^n 2^i$ im 6 dimensionalen Raum ungünstig und würde für hohe n den Rahmen bezüglich des Rechenaufwands sprengen, da für jeden Spot jeweils zusätzlich noch 100 Nachbarn bestimmt werden müssen. Für kleine n wäre der Rechenaufwand mit 100 Nachbarpunkten pro Spot zwar überschaubar, aber die Verteilung der Spots wäre viel zu grob, sodass der Umweg über die Zufallsvektoren gegangen werden muss, was aber mit der Betrachtung der Nachbarpunkte im sphärischen Radius dennoch zu einem guten Betrachtungsspektrum führt. Es werden für diese Untersuchung insgesamt 10000 Spots bestimmt, also werden insgesamt $10000 \cdot 100 = 1000000$ Stablängenkonfigurationen in die Betrachtung miteinbezogen. Die Wahrscheinlichkeit einer Clusterung der Zufallswerte ist bei der verhältnismäßig kleinen Anzahl an bestimmten Spots im Suchraum $[0.01\text{m}, 1.00\text{m}]^6$ gering und der Zufall bietet den Vorteil, dass Spots getroffen werden, die mit Hilfe der Gitterstruktur der *van der Corput-Sequenz* nicht getroffen werden und so eine bessere Aussage bezüglich des Steigungsverhaltens gemacht werden kann. Die Regelmäßigkeit der Messung wie in Abschnitt 4.3 kann zu Lücken innerhalb der Fitness Cloud führen, in denen keine Aussage über das Steigungsverhalten getroffen werden kann. Folglich ist die Betrachtung über Zufallsvektoren für die Fitness Cloud verstärkt von Vorteil.

4.4.3 Ergebniss

In diesem Abschnitt folgt die Darstellung der Fitness Cloud zu der bekannten Zielfunktion. Die FC wird erläutert und anschließend näher untersucht.

Anhand des Plots 4.4 lassen sich einige für die Evaluation relevante Aussagen treffen. Als erstes fällt auf, dass die Durchschnittslinie der Fitnesswerte linear und zusätzlich auch auf der Ursprungsgerade verläuft. Das bedeutet, dass sich die Fitness eines betrachteten Nachbars im Durschnitt nicht signifikant unterscheiden wird. Folglich kann von einer durchschnittlich relativ

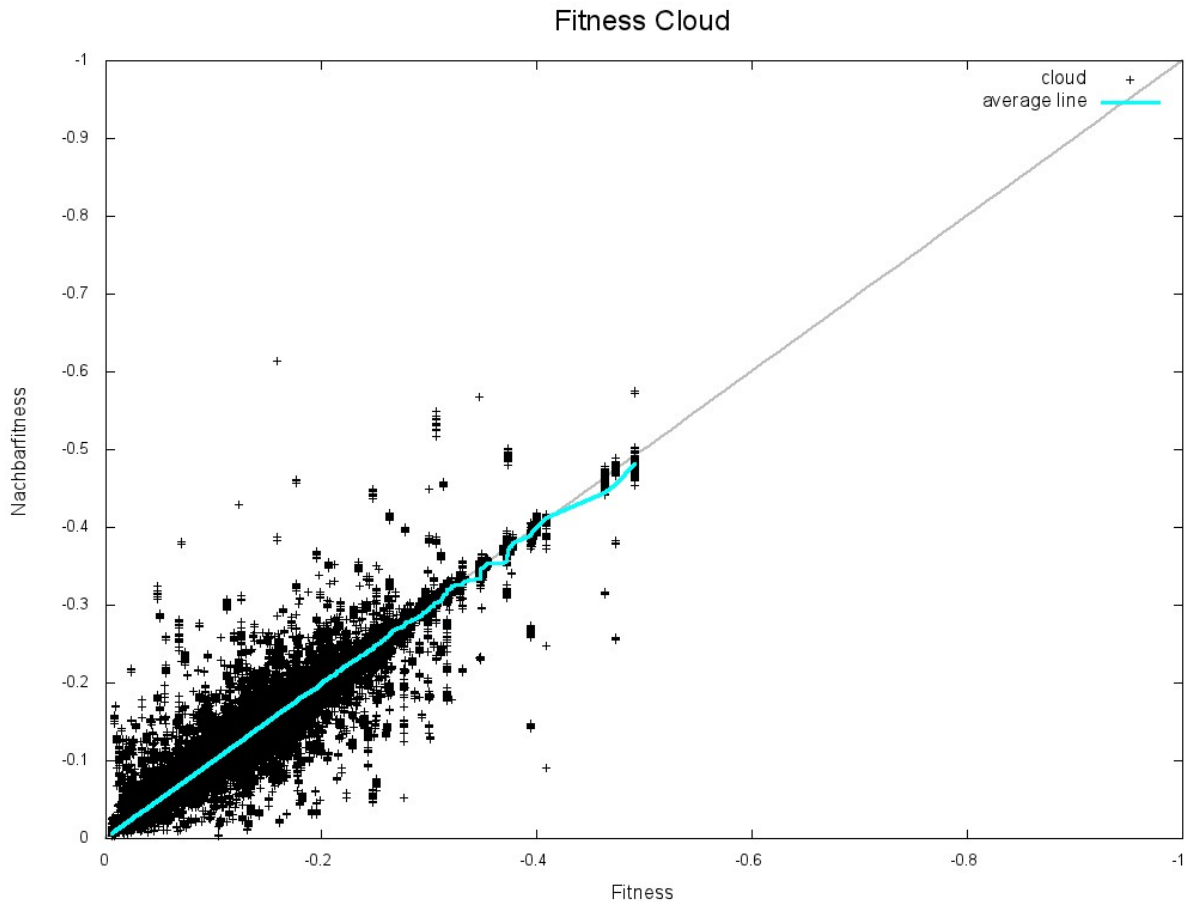


Abbildung 4.4: Plot der Fitness Cloud

flachen Steigung der Fitnesslandschaft gesprochen werden. Als nächstes fällt auf, dass sich die gemessenen Werte eher zur Fitness 0 statt zur Fitness -1 ansammeln. Also ist die allgemeine Wahrscheinlichkeit Fitnesswerte in Richtung -1 zu treffen verhältnismäßig gering, sodass eher mit Werten in Richtung der Fitness 0 gerechnet werden muss. Im Kontext lässt sich sagen, dass die Fitnesswerte in Richtung der Fitness von -1 also Stellen sind, an denen die Steigung plötzlich sehr stark wird, also in schmalen, aber auch seltenen Stellen liegen. Wie zu sehen ist existieren viele Ausreißer innerhalb der Fitness Cloud, die sehr stark von dem Durchschnitt abweichen. Wenn die dazugehörigen Parameter für die Ausreißer ausgegeben werden, werden unter anderem folgende Parameter ersichtlich:

Auffällig ist, dass die dazugehörigen Parameter in hoher Schräglage sind, was schon in Tabelle 4.2 des modifizierten Hill-Climber Algorithmus zu sehen war. Dies lässt die Vermutung entstehen: Je stärker die Schräglage ausgeprägt ist, desto höher ist die Abweichung von der

$l_{11...21}$	$l_{12...22}$	$l_{13...23}$	$Fitness$	$Fitness_{Spot}$
0.106	0.343	0.455	-0.278	-0.152
0.349	0.506	0.019		
0.487	0.863	0.267	-0.199	-0.125
0.110	0.641	0.209		
0.592	0.019	0.683	-0.181	-0.044
0.216	0.432	0.847		
0.827	0.332	0.278	-0.068	-0.110
0.862	0.045	0.165		
0.506	0.806	0.059	-0.137	-0.220
0.103	0.655	0.697		

Tabelle 4.3: Exemplarische Parameter und Fitnesswerte der Ausreißer in der Fitness Cloud

Durchschnittsfitness. Dies lässt sich dadurch erklären, dass der Arbeitsraum der Roboters bei starker Schräglage verhältnismäßig klein ist. Dies hat zur Folge, dass die, mit Hilfe der Van-Der-Corput-Sequenz, Anzahl gemessener, valider Posen zur Bestimmung des GCI einer Stablängenkonfiguration zu gering ist und so die Stablängenkonfiguration entweder zu gut oder zu schlecht bewertet wird. Somit ist eine Untersuchung der Verkleinerung des Suchraums $[0.01\text{m}, 1.00\text{m}]^6$ unumgänglich um Schräglagen zu vermeiden.

4.4.4 Untersuchung zur Suchraumgröße

Für die Bestimmung eines gut gewählten Suchraums ist eine systematische Vorgehensweise von Nöten. Als erstes werden Stablängenkonfigurationen im Suchraum $[0.4\text{m}, 0.5\text{m}]^6$ untersucht, was sich an dem Datenblatt des FlexPickers von *ABB Robotics* [18] orientiert, und inkrementell (0.1m) immer weiter in beide Richtungen erweitert, aber ohne den zuvor untersuchten Suchraum. Zur Veranschaulichung der Systematik dient Abbildung 4.5.

Diese Untersuchung gibt Auskunft darüber, ab welchen Intervallsgrößen Ausreißer vermehrt vorkommen. Ist ein Suchraum bekannt in dem wenig bis keine Ausreißer vorkommen, kann von einer Fitnesslandschaft ohne verhältnismäßig großen, plötzlichen Steigungen gesprochen werden.

Es folgen die einzelnen Plots für die einzelnen Suchräume I, II, III und eine Gesamtansicht über alle Teilsuchräume, also über den Suchraum $[0.2\text{m}, 0.7\text{m}]^6$ in Abbildung 4.6. Wie zu erkennen ist, fängt es erst bei der Betrachtung des Suchraums III an zu streuen. Nun wird untersucht, wie sich die Streuung innerhalb der Suchraums III verhält. So werden jetzt die



Abbildung 4.5: Übersicht der zu untersuchenden Teilsuchräume zur Analyse des geeigneten Suchraums

Suchräume $[0.25\text{m}, 0.3\text{m}]^6 \wedge [0.6\text{m}, 0.65\text{m}]^6$ und $[0.2\text{m}, 0.25\text{m}]^6 \wedge [0.65\text{m}, 0.7\text{m}]^6$ untersucht. Die dazugehörigen Fitness Clouds sind auf Abbildung 4.7 zu sehen.

Schon in der ersten Fitness Cloud in Abbildung 4.7 ist zu sehen, dass es hier zu streuen beginnt und folglich die Auswirkungen der Stablängenkonfiguration in Schräglage ihre Auswirkungen haben. Eine Erhöhung führt immer weiter zu stärkeren Streuungen, aber auch zu einer Verlängerung der Fitness Cloud anlang der Ursprungsgerade. Dies lässt sich wieder dadurch erklären, dass der Arbeitsraum verhältnismäßig klein ist und dadurch, wie die Untersuchung es zeigt, schon im Schnitt bessere Fitnesswerte ergibt, aber mit der Konsequenz, dass vermehrt Ausreißer vorkommen. So liegt hier der Fokus darauf, dass weniger Ausreißer vorliegen, um unvorhergesehene bzw. durch Zufall zu gute oder schlechte Ergebnissentwicklungen zu vermeiden. Anhand von Abbildung 4.6 ist zu erkennen, dass der Suchraum $[0.3\text{m}, 0.6\text{m}]^6$ die geringste Streuung aufweist und folglich für die Evaluation gewählt werden sollte um starke Sprünge innerhalb der Fitnesslandschaft zu vermeiden.

4.5 Zusammenfassung

Es folgt eine kurze Übersicht der ermittelten Eigenschaften der Zielfunktion:

- Die 6 dimensionale Zielfunktion ist nicht separierbar und behält dadurch ihre Komplexität.
- Die Laufzeit ist abhängig von der Größe der *van der Corput-Sequenz* und der Dimensionierung der Jacobimatrix.
- Es existiert keine Korrelation und die Zielfunktion ist multimodal.
- Die Steigungen sind im Durchschnitt überschaubar, aber es existieren Ausreißer.
- Ausreißer zeigen gute wie auch schlechte Lösungen an den Randgebieten, sodass der Suchraum reduziert werden muss um Ausreißer zu vermeiden.

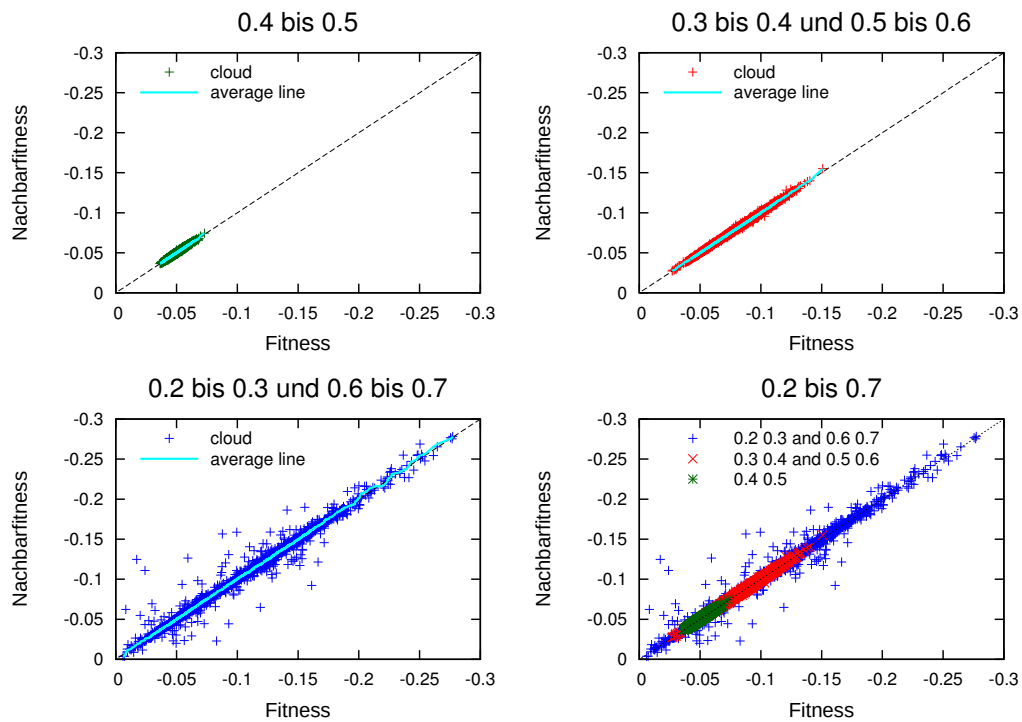


Abbildung 4.6: Fitness Clouds der untersuchten Teilsuchräume zur Analyse des geeigneten Suchraums

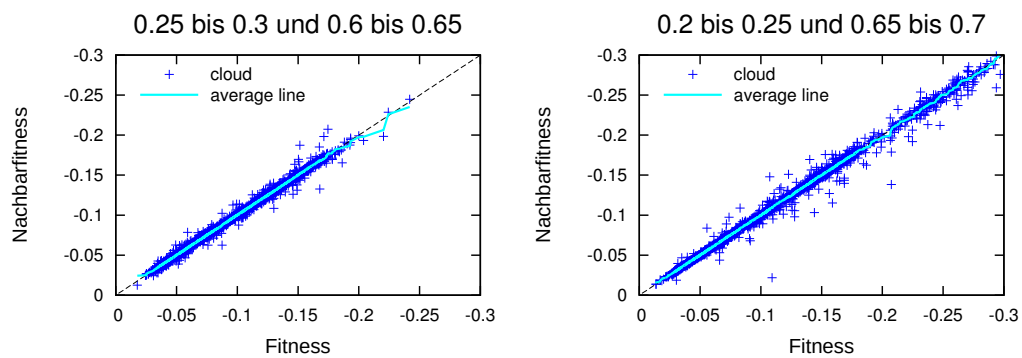


Abbildung 4.7: Fitness Clouds des aufgespaltenen streuenden Teilsuchraums zur Analyse des geeigneten Suchraums

5 Evaluation des Standard Particle Swarm Optimization 2011

Übersicht

5.1	Systematik der Analyse	31
5.2	Ausgangssituation	32
5.3	Parameteranalyse	33
5.4	Vergleich	39

Der SPSO-11 ist ein Schwarm-Algorithmus, der aus einer Menge von Partikeln besteht, die in einem Suchraum mit n Dimensionen nach einer optimalen Lösung suchen. Zur kurzen beispielhaften Veranschaulichung des Verhaltens von Partikelschwärmen dient die folgende Abbildung 5.1.

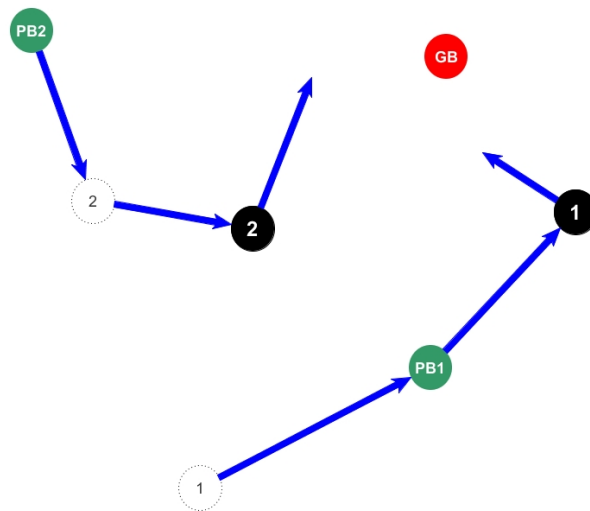


Abbildung 5.1: Beispielhafte Veranschaulichung des Verhaltens von Partikeln

Jede Bewegung eines Partikels im Suchraum ist, abhängig von der eigenen Beschleunigung (blaue Pfeile). Die Beschleunigung wird unter Anderem von der (lokalen) Anziehungskraft der partikel bezogenen besten Lösung (grüne Punkte) und der (globalen) Anziehungskraft der insgesamt besten gefunden Lösung (roter Punkt) beeinflusst. In der Abbildung 5.1 sind zwei Partikel zu sehen. Partikel 1 hatte eine erste Bewegung gemacht und daraufhin eine bessere eigene Lösung gefunden. In den folgenden Schritten wird die Suchrichtung und -weite von den lokal und global besten Punkten beeinflusst, sodass Partikel 1 in Richtung der global besten Lösung

geht, aber dennoch unter Einfluss der lokalen Anziehungskraft zu Punkt PB1 steht. Partikel 2 hat von Anfang eine beste lokale Lösung gefunden. Wenn davon ausgegangen wird, dass bei der Suche keine bessere lokale Lösung gefunden wird, wird die Bewegung von Partikel 2 von anfang an von der eigenen lokalen besten wie auch von der globalen Lösung angezogen. Daran kann gesehen werden, dass eine Gewichtung der jeweiligen Anziehungskräfte von nöten ist, damit Partikel zu einer Lösung konvergieren können. Desweiteren kann die Ausprägung der entstehenden Beschleunigung der jeweiligen Partikel erhalten, gedrosselt oder erhöht werden um das Verhalten der Partikel zu beeinflussen.

Es sind im vorhinein Parametrisierungen durchzuführen, die das Verhalten des Algorithmus beeinflussen. Unter anderem muss die Populationsgröße S , also die Menge an Partikeln, festgelegt werden. Jeder dieser Partikel bewegt sich unter dem Einfluss der eigenen Beschleunigung $\vec{v}_i$ und weiterer Parameter durch den Suchraum. Die weiteren Parameter, die in dieser Arbeit behandelt werden, sind die Beschleunigungskonstante w , die lokale Anziehungskraft c_1 und die globale Anziehungskraft c_2 .

Die Besonderheit des SPSO-11 ist, dass zusätzlich eine Hypersphere H_i mit einem Gravitationspunkt $\vec{G}_i$ als Mittelpunkt mit Radius $\|\vec{G}_i - \vec{x}_i\|$ entsteht. $\vec{G}_i$ entsteht im Bezug der aktuellen Position des Partikels $\vec{x}_i$, der partikelbezogenen Position $\vec{p}_{best_i}$ mit dem partikelbezogenem besten Wert und der globalen Position $\vec{g}_{best_i}$ des global besten Werts. Aus diesen Positionen entstehen $\vec{p}_i$ als Bewegungsrichtung zur partikelbezogenen besten Position vom jeweiligen Partikel aus und $\vec{g}_i$ als Bewegungsrichtung zur global besten Position vom jeweiligen Partikel aus.

Innerhalb von H_i wird ein uniform verteilter Zufallsvektor $\vec{h}_i$ gebildet von dem aus die Bewegung des Partikels mit vorhandener Beschleunigung weitergeht. Die folgende formale Darstellung bezieht sich auf das Verhalten eines Rechenschritts. Der Verweis (t) ist für genau diesen Rechenschritt t jeweils bei den Variablen mit angegeben:

$$\vec{p}_i^{(t)} = \vec{x}_i^{(t)} + c_1 \vec{R}_1^{(t)} \vec{p}_{best_i}, \quad \vec{g}_i^{(t)} = \vec{x}_i^{(t)} + c_2 \vec{R}_2^{(t)} \vec{g}_{best_i}$$

$$\vec{G}_i^{(t)} = \begin{cases} \frac{\vec{x}_i^{(t)} + \vec{p}_i^{(t)} + \vec{g}_i^{(t)}}{3}, & \text{falls } \vec{p}_{best_i} \neq \vec{g}_{best_i} \\ \frac{\vec{x}_i^{(t)} + \vec{p}_i^{(t)}}{2}, & \text{falls } \vec{p}_{best_i} = \vec{g}_{best_i} \end{cases}$$

$$\vec{x}_i^{(t+1)} = w \vec{v}_i^{(t)} + \vec{h}_i^{(t)}, \quad \vec{v}_i^{(t+1)} = w \vec{v}_i^{(t)} + \vec{h}_i^{(t)} - \vec{x}_i^{(t)}, \quad \vec{h}_i^{(t)} \in H_i(\vec{G}_i^{(t)}, \|\vec{G}_i^{(t)} - \vec{x}_i^{(t)}\|)$$

mit $\vec{R}_1^{(t)}$ und $\vec{R}_2^{(t)}$ als uniform verteilte Zufallsvektoren aus $[0, 1]^n$. Die Systematik mit der Hypershare und dem dazugehörigen Gravitationspunkt verspricht eine bessere und weitreichendere Suche als in Vorgängerversionen des SPSO [8][20].

Die experimentellen Ergebnisse ergaben, dass die Zielfunktion eine multimodale Funktion mit im Durchschnitt mittelmäßiger Steigung ist. Es wäre zu erwarten, dass die Beschleunigungskonstante w hoch angesetzt werden soll, um aufgrund der multimodalität der Zielfunktion das explorative Verhalten des Schwarm zu erhöhen. Die *Fitness Cloud* deutet nämlich darauf hin, dass in der Nähe guter Lösungen auch immer bessere Lösungen gefunden werden können, da die Mittelwertlinie der *Fitness Cloud* entlang der Ursprungsgerade verläuft. Desweiteren ist zu erwarten, dass die lokale Anziehungskraft c_1 höher gewichtet werden soll als die globale Anziehungskraft c_2 um die Bewegung eines Partikel durch Bewegungen von Partikeln in irreführenden Steigungen nicht negativ zu beeinflussen und so die Möglichkeit besteht mehr (ggf. bessere) Optima abzusuchen. Unter irreführender Steigung ist gemeint, dass sich Partikel in einem Bereich befinden, wo die Steigung zu einem Optimum so stark ist, dass schnell eine bessere Lösung gefunden wird und geringere Steigungen in der Umgebung folglich ignoriert werden, die aber auf ein noch besseres Optimum laufen. Über die Populationsgröße S lässt sich vorerst nichts aussagen, sodass ein Vergleich zwischen verschiedenen Populationsgrößen durchgeführt werden muss.

In diesem Kapitel wird analysiert, ob sich der *Standard Particle Swarm Optimization 2011* Algorithmus (SPSO-11) für die Zielfunktion dieser Arbeit eignet. Die Eignung soll anhand eines Vergleichs des SPSO-11 mit drei parameterfreien Algorithmen geschehen. Die in Bezug genommenen Algorithmen sind die *Zufallssuche*, der *klassische Hill Climber* und der *Hooke-Algorithmus*.

Bei der *Zufallssuche* wird in jedem Evaluationsschritt für jede Dimension der Zielfunktion ein uniform verteilter Zufallswert aus dem gültigen Wertebereich, hier $[0.3m, 0.6m]$, bestimmt und die dazugehörige Lösung berechnet. Der Algorithmus endet nach einer festen Anzahl an Evaluationsschritten oder nach Findung einer Lösung, die eine qualitative Schranke überschritten hat.

Der *Hill Climber* wird mit einem uniform verteilten Startpunkt im Suchraum initialisiert. Bei jedem Evaluationsschritt wird in einem konstanten sphärischen Radius um die aktuelle Position herum uniform verteilt nach einer besseren Lösung gesucht. Ist eine bessere Lösung gefunden, ändert der Algorithmus die aktuelle Position auf die bessere Lösung und die Suche geht von der neuen Position aus genau so weiter. Der Algorithmus hat die selben Abbruchkriterien wie die *Zufallssuche*.

Beim *Hooke-Algorithmus* wird ebenfalls so wie beim *Hill Climber* initialisiert. Zusätzlich beginnt der *Hooke-Algorithmus* mit einer anfangs festen Suchweite in alle positive und negative kartesische Richtungen. Alle Lösungswerte der Positionen in Suchweite der aktuellen Position werden mit der Lösung der aktuellen Position verglichen. Ist einer dieser Positionen in Suchweite besser, ändert der *Hooke-Algorithmus* seine Position und schau in der gleichen Suchweite nach neuen besseren Lösungen. Ist keine der potenziellen Lösungen in Suchweite besser, wird die Suchweite dauerhaft verkleinert und der Algorithmus beginnt mit der Prüfung der Lösungen in neuer Suchweite. Die Abbruchkriterien für den *Hooke-Algorithmus* sind die Unterschreitung einer bestimmten Suchweite bzw. das Erreichen der Schrittweite 0 oder die schon oben erwähnten Kriterien.

Der Aufbau der Evaluation sieht wie folgt aus: Als Erstes wird auf die gererelle Systematik der Parameteranalyse eingegangen um nachfolgende Untersuchungen besser verstehen zu können. Der Start der Analyse erfolgt durch die Bestimmung einer Ausgangssituation. Diese Ausgangssituation wird zeigen, mit wie vielen Rechenschritten die jeweiligen Algorithmen ein bestimmtes Ergebniss erreichen werden, sodass damit schon ein direkter Vergleich zu sehen sein wird. Nach der Kenntnissnahme der Ausgangssituation folgt eine Parameteranalyse für den SPSO-11. Mit der ermittelten Parametrisierung folgt ein Vergleich mit den o.g. drei Algorithmen um abschließend sagen zu können, ob der SPSO-11 ein geeigneter Algorithmus für die Zielfunktion ist.

5.1 Systematik der Analyse

Für die Analyse der Parameterfaktoren des SPSO-11 w , c_1 und c_2 bietet es sich an die Faktoren in Faktorstufen zu unterteilen und jede Faktorkombination auf ihre Effizienz zu prüfen [19]. Unter Effizienz bei algorithmischen Vergleichen versteht man den direkten Vergleich von benötigten Evaluationsschritten zum Finden einer festgelegten Lösung. Konkrete Laufzeiten zu vergleichen ist nicht von Vorteil, da sich die Laufzeit je nach Architektur der Recheneinheit sehr stark unterscheiden kann, sodass die Betrachtung von Evaluationsschritten als angemessen eingestuft werden kann. Die Untersuchung des Einflusses der Parametrisierung auf die Evaluationsschritte ist der Kernaspekt der Betrachtung der Effizienz eines parametrisierbaren Algorithmus [5].

Die in dieser Arbeit insgesamt betrachteten Faktorstufen sind folgende:

Es entstehen also folglich insgesamt $3^9 = 19683$ Faktorstufenkombinationen und für jede einzelne Faktorstufe insgesamt $2^9 = 512$ mögliche Kombinationspartner, was ein breites Spektrum der Betrachtung mit sich bringt. Die möglichen Kombinationen werden im Abschnitt 5.3.1 näher untersucht.

Faktorstufe	0	1	2	3	4	5	6	7	8
w	0.5	0.6	0.7	0.8	0.9	1.0	1.1	1.2	1.3
c_1, c_2	$\frac{1}{16}$	$\frac{1}{8}$	$\frac{1}{4}$	$\frac{1}{2}$	1	2	3	4	5

Tabelle 5.1: Übersicht der zu untersuchenden Faktorstufen zur Evaluation des SPSO-11

In den Ausarbeitungen von Clerc [8][20] wird auch auf die Populationsgröße eingegangen. Es wird gesagt, dass sich beim SPSO-11 an einer Populationsgröße $S = 40$ angelehnt werden soll, wobei dieser Wert je nach Situation angepasst werden muss. In dieser Arbeit werden Populationsgrößen innerhalb von 2 bis 40 Partikeln untersucht. Eine Partikelgröße von 1 hat keinen Zweck, da sonst ein *Hill Climber* ähnlicher Algorithmus entsteht. Wie sich S auf die Effizienz des SPSO-11 auswirkt wird der Abschnitt 5.3.2 zeigen.

Anhand einiger Messungen der parameterfreien Algorithmen zeigte sich, dass die zu erreichende Schranke der Fitness -0.13 betragen sollte. Diese Schranke könnte im Verhältnis als nicht zu schwer, aber auch als nicht zu einfach zu erreichen eingestuft werden. Das Ziel dieses Kapitels ist folglich den SPSO-11 so zu parametrisieren, dass die Schranke von -0.13 im Idealfall nach weniger Evaluationsschritten erreicht wird als mit den parameterfreien Algorithmen.

5.2 Ausgangssituation

Die Ausgangssituation spiegelt wieder wie gut die o.g. drei Algorithmen zu dem SPSO-11 stehen. Die erste Betrachtung des SPSO-11 im Vergleich wird mit den Standardparametrisierungen von Clerc [20] durchgeführt. Die Standardparametrisierungen sind: $S = 40$, $w = \frac{1}{2 \cdot \ln(2)}$ und $c_1 = c_2 = \frac{1}{2} + \ln(2)$. Es werden für jeden Algorithmus jeweils 100 mal 500 Evaluationsschritte ausgeführt um am Ende zu sehen wie hoch die durchschnittliche Fitness, die Standardabweichung und das Minimum wie auch Maximum nach jeweils 500 Evaluationsschritten ist. Die Abbildung 5.2 zeigt die jeweiligen Ergebnisse. Es ist zu erkennen, dass der SPSO-11 mit den Standardparametern und einer im Schnitt benötigten Anzahl an Evaluationsschritten von 274 im Verhältnis sehr schlecht abschneidet. Da es sich hier aber nur um eine potenzielle Parametrisierung gehandelt hat, sind die Möglichkeiten des SPSO-11 noch längst nicht ausgeschöpft, sodass eine Parameteranalyse durchgeführt werden muss um zu prüfen, ob sich der SPSO-11 unter geeigneten Parametern als effizient zeigen wird.

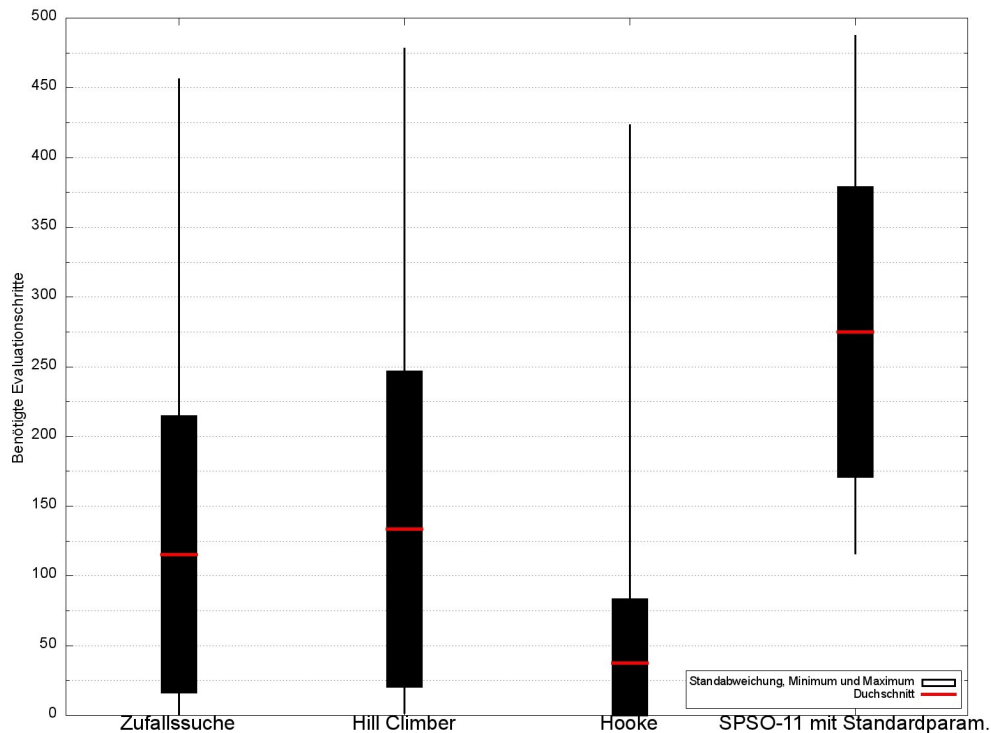


Abbildung 5.2: Übersicht der gefundenen Lösungen der zu vergleichenden parameterfreien Algorithmen und dem SPSO-11 mit Standardparametrisierung

5.3 Parameteranalyse

5.3.1 Faktoranalyse

In der Faktoranalyse dieser Arbeit werden die zu untersuchende Parametrisierungen aus der Tabelle 5.1 entnommen. Für die Faktoranalyse wird eine Populationsgröße $S = 40$ genutzt. Jede der Parametrisierungen wird 10 mal mit maximal 500 Evaluationsschritten ausgeführt um im Durchschnitt einen Trend erkennen zu können ab welchem Evaluationsschritt die Fitness von -0.13 überschritten wird. Es werden zuerst die Faktorstufen 0 bis 5 evaluiert: $w = \{0.5, 0.6, \dots, 1.0\}$ werden zeigen wie gut oder schlecht der Algorithmus auf Drosselungen bzw. Erhaltung der Eigenbeschleunigung reagiert. $c_1 = c_2 = \{\frac{1}{16}, \frac{1}{8}, \frac{1}{4}, \dots, 2\}$ werden herausstellen, wie wenig relevant die jeweiligen Anziehungskräfte zu sein scheinen. $c_1 = c_2 = 2$ würde durch den Einfluss des Zufallsvektors bedeuten, dass die Anziehungskraft sich im Schnitt für die jeweilige Dimension nicht ändert.

In Abbildung 5.3 ist eine Übersicht von 6 Gittern zu sehen, die jeweils für eine Faktorstufe für w stehen und die jeweiligen Achsen repräsentieren die Parameter c_1 und c_2 . Die Parametrisierung mit der geringsten Anzahl an Evaluationsschritten ist hier $w = 1.0$, $c_1 = \frac{1}{8}$ und $c_2 = 2$ mit

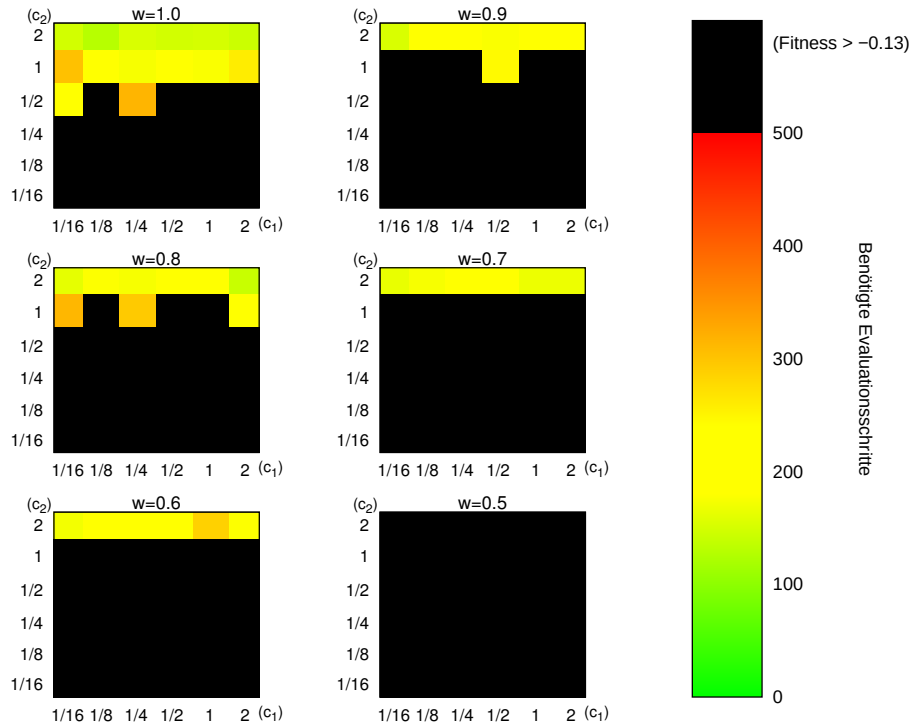


Abbildung 5.3: Gitterübersicht zur benötigten Anzahl an Evaluationschritten zur Grenzüberschreitung mit niedrigen Faktorstufen und Populationsgröße 40

129 Evaluationsschritten. Wie zu sehen ist, sind zwei grundlegende Trends zu erkennen: Die Beschleunigungskonstante w soll folglich höher angesetzt werden, da mit steigendem w die Anzahl und auch Qualität der Evaluationsdurchläufe besser wird. Das selbe gilt ebenfalls mit steigender globaler Anziehungskraft c_2 . Über die lokale Anziehungskraft c_1 lässt sich keine Aussage treffen, da sich auch bei genauer Betrachtung der konkreten Werte kein Verlauf und keine Konvergenz zeigt. Folglich ist der Parameter c_1 irrelevant für die Effizienz des SPSO-11 für die betrachtete Zielfunktion. Da w und c_2 den Trend aufweisen, dass ein höherer Wert überhaupt bzw. schneller zum Ergebniss führt, ist eine Betrachtung der Faktorstufen 6 bis 9 nötig. Für c_1 kann aufgrund der Irrelevanz ein beliebiger Wert gewählt werden.

Für die Betrachtungen der höheren Faktorstufen wird die lokale Anziehungskraft c_1 auf 1 gesetzt um den Zufallsvektor $\vec{R} \in [0.1]^n$ nicht zu beeinflussen, was aber im Wesentlichen, wie oben zu sehen, sowieso keine Auswirkungen hat. Die betrachteten Faktorstufen sind nun 3 bis 8, also sind die untersuchten Werte $w = \{0.8, 0.9, \dots, 1.3\}$ und $c_2 = \{\frac{1}{2}, 1, 2, \dots, 5\}$. Diese Werte werden mit der selben Anzahl an Durchgängen und maximalen Evaluationsschritten wie oben evaluiert. In Abbildung 5.4 ist ein Gitter zu den eben angegebenen Faktorstufen und

$c_1 = 1$ zu sehen. Die beste Parametrisierung ist hier $w = 1.1$, $c_1 = 1$ und $c_2 = 5$ mit 94

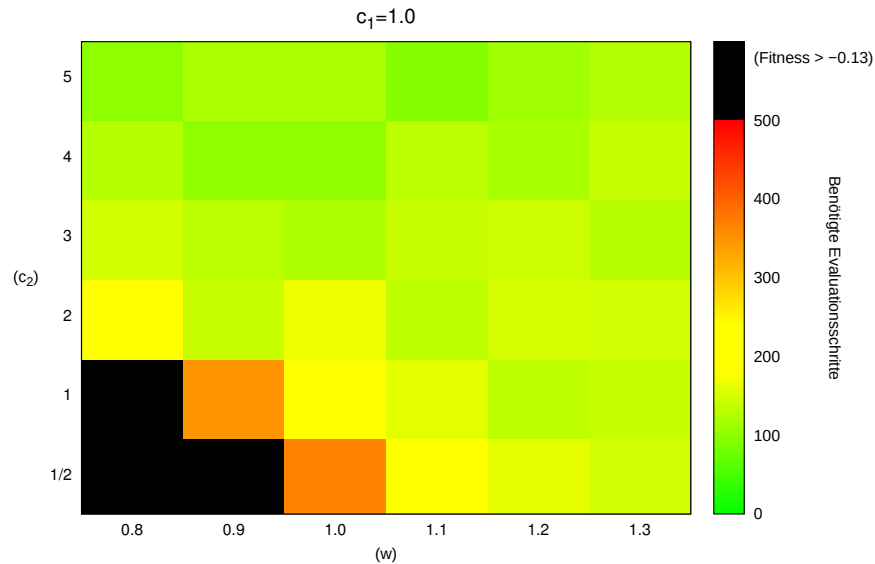


Abbildung 5.4: Gitterübersicht zur benötigten Anzahl an Evaluationschritten zur Grenzüberschreitung mit hohen Faktorstufen und Populationsgröße 40

Evaluationsschritten. Auch hier ist der selbe Trend bezüglich w und c_2 zu erkennen. Für höhere Parametereinstellungen als in Tabelle 5.1 abgegeben, ist kein Trend für eine weitere Verringerung der Evaluationsschritte bezüglich w oder c_2 erkennbar.

Ein anderer interessanter Aspekt im Bezug auf die Parametrisierung ist, wie gut die gefundenen Lösungen werden, wenn keine Grenze des Fitnesswertes gesetzt wird und jeder Durchgang 500 Evaluationen durchläuft. Dadurch kann geprüft werden, ob sich im Bezug der Qualität der Lösung ohne Grenzsetzung ebenfalls ein Trend zeigt. Die Betrachtung mit den niedrigen Faktorstufen 0 bis 5 kann aus Abbildung 5.5 entnommen werden. Die beste Lösung, die jeweils im Durchschnitt nach 500 Evaluationsschritten gefunden wurde ist -0.181 mit den Parametern $w = 1.0$, $c_1 =$ und $c_2 = 2$. Auch bei der Betrachtung ermittelten Fitness nach der maximalen Anzahl an Evaluationsschritten stellt sich der gleiche Trend dar. Ein hohes w und c_2 steht für eine bessere Fitness und der Faktor c_1 zeigt sich auch bei dieser Betrachtung als irrelevant. Für die Faktorstufen 3 bis 8 mit $c_1 = 1$ wurde als beste Fitness mit den Parametern $w = 1.2$ und und $c_2 = 5$ der Wert -0.189 erreicht. Es ist auch hier der gleiche Trend analog zu Abbildung 5.4 vorhanden.

Der Verlauf der Parameterbetrachtungen widerspricht der Erwartung am Anfang des Kapitels

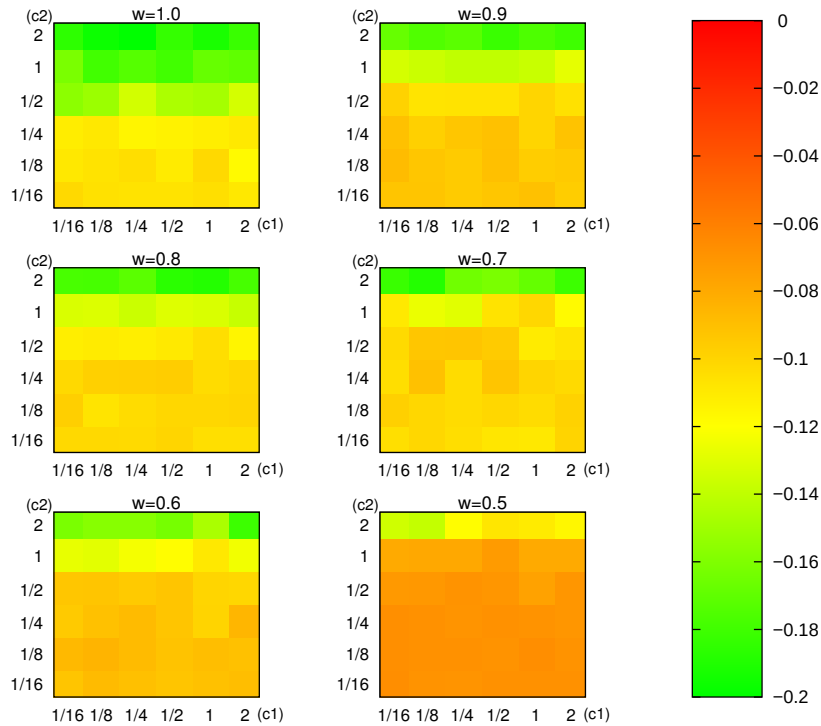


Abbildung 5.5: Gitterübersicht zu Qualität der Lösung nach maximaler Anzahl an Evaluationschritten mit niedrigen Faktorstufen und Populationsgröße 40

bezüglich c_1 und c_2 , wobei sich die Erwartung gegenüber w erfüllen. Bei der Betrachtung der gefundenen Lösungen im Suchraums fällt auf, dass sich die Partikel zu den Randgebieten des gewählten Suchraums $[0.3, 0.6]^6$ bewegen. Die Werte an den Randgebieten, die gefunden wurden, tendieren zur Schiefelage. Dies ergänzt die Untersuchung der Suchraumgröße im Abschnitt 4.4.4. Der anfängliche Trend für w und c_2 ist nach den Ergebnissen und dem Vorwissen über die Zielfunktion im Nachhinein eigentlich keine große Überraschung. Die Partikel beeinflussen sich gegenseitig über die globale Anziehungskraft c_2 so, dass die Lösungen im Suchraum in Schiefelage an Dominanz gewinnen und die lokale Anziehungskraft c_1 an Bedeutung verliert.

Somit kann geschlussfolgert werden, dass die Parametrisierung der Faktoren mit höheren Faktorstufen keine Änderung beim Verhalten des Schwarms haben wird, da die Randgebiete des Suchraum, die eine bessere Fitness hervorbringen, präferiert werden und verhältnismäßig schnell gefunden werden. Die relativ zu den verglichenen Algorithmen in Abschnitt 5.2 hohe Anzahl an Evaluationsschritten lässt sich durch die Populationsgröße $S = 40$ erklären. Für jeden dieser Partikel muss einzeln eine Rechnung erfolgen, was in vielen Fällen des Evaluationsverlaufs unnötig ist, da andere Partikel schon eine bessere Lösung gefunden haben. Somit ist eine

Untersuchung der Populationsgröße S von Nöten um die Effizienz des SPSO-11 steigern zu können.

5.3.2 Populationsgrößenanalyse

Es folgt eine Betrachtung der Faktorstufen 0 bis 5. Innerhalb dieses Spektrums werden wieder jeweils 10 mal maximal 500 Evaluationsschritte durchgeführt. Es wird untersucht mit welcher Populationsgröße S die geringste Zahl an Evaluationsschritten zur Überschreitung der Grenze von -0.13 hervorgeht und wie gut der beste Fitnesswert ohne Grenzsetzung jeweils ist.

Wie oben erwähnt ist die Standardeinstellung für den SPSO-11 nach Clerc [8] bezüglich der Populationsgröße $S = 40$. In seiner Ausarbeitung wurde ebenfalls für frühere Versionen des SPSO von folgender Formel für die Populationsgröße in einem n -dimensionalen Suchraum gesprochen:

$$S = 10 + \lfloor 2\sqrt{n} \rfloor \xrightarrow{n=6} S = \lfloor 14,899 \rfloor = 14 \quad (5.1)$$

Diese Formel wird ebenfalls auf ihre Tauglichkeit überprüft. Um eine Orientierung über die Populationsgrößen und ihre Effizienz zu gewinnen, wurden die Populationen $S_1 = \{5, 10, 15, 20, 40\}$ und darauf $S_2 = \{2, 3, \dots, 9, 10\}$ anhand der o.g. Aspekte untersucht und in Abbildung 5.6 dargestellt. In Betrachtung der minimal benötigten Evaluationsschritte zeigte sich hier vorerst der Trend, dass eine geringe Anzahl an Partikeln besser sei. Es zeigt sich auch, dass die Standardeinstellung $S = 40$ und Clercs Formel (5.1), die $S = 14$ ergab, unzutreffend bzgl. der betrachteten Zielfunktion ist. Auf Basis des Trends in Abbildung 5.6 ergab sich die Untersuchung anhand der Populationsgrößen S_2 , dessen Ergebniss in Abbildung 5.7 zu sehen ist. Das Minimum an benötigten Evaluationsschritten liegt insgesamt bei $S = 5$ mit 37 Evaluationsschritten. Daran ist zu sehen, wie wichtig die Wahl einer geeigneten Populationsgröße ist. Die gefundene Populationsgröße mit den unteren Faktorstufen lässt gegenüber der Standardeinstellung $S = 40$ eine Reduzierung der Evaluationsschritte auf $\frac{37}{129} \Rightarrow 28.68\%$ zu. Zur Veranschaulichung höherer Faktorstufen dient die Gitteransicht für $c_1 = 1$ und $S = 5$ in Abbildung 5.8. Die geringste benötigte Anzahl an Evaluationsschritten ist 27 und wurde mit den Parametern $w = 1.3$ und $c_2 = 5$ erreicht. Zum Einen ist hier zu erkennen, dass sich die Aussagen für die Parameterfaktoren w , c_1 und c_2 aus dem vorherigen Abschnitt auch für $S = 5$ zeigen. Untersuchungen anderer Populationsgrößen zwischen 2 und 40 ergaben ebenfalls die selben Aussagen für die Parameterfaktoren, sodass die Aussagen allgemeingültig für die Zielfunktion sind.

Bezüglich der Qualität der Lösung ohne Grenzsetzung nach 500 Evaluationsschritten ist der Trend erkennbar, dass höhere Populationsgrößen im Schnitt eine bessere Fitness ergeben. Dies lässt sich dadurch erklären, dass bei hoher Populationsgröße, die Wahrscheinlichkeit höher ist

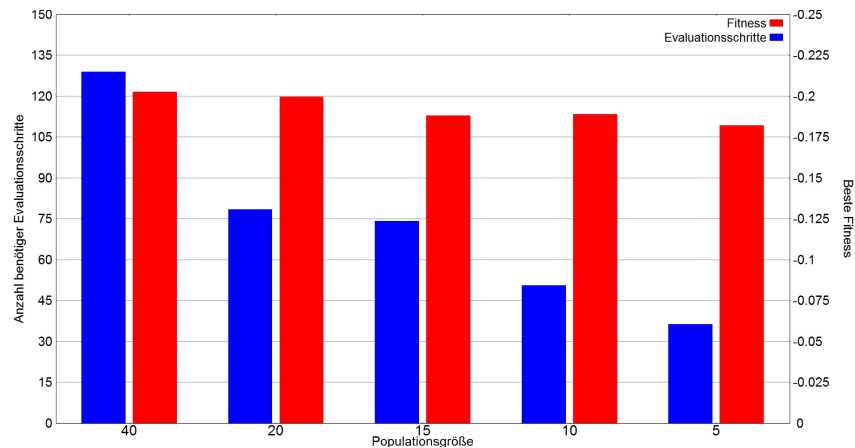


Abbildung 5.6: Geringste Anzahl benötigter Evaluationsschritte und beste gefundene Fitness für Populationen der Größen 5, 10, 15, 20 und 40

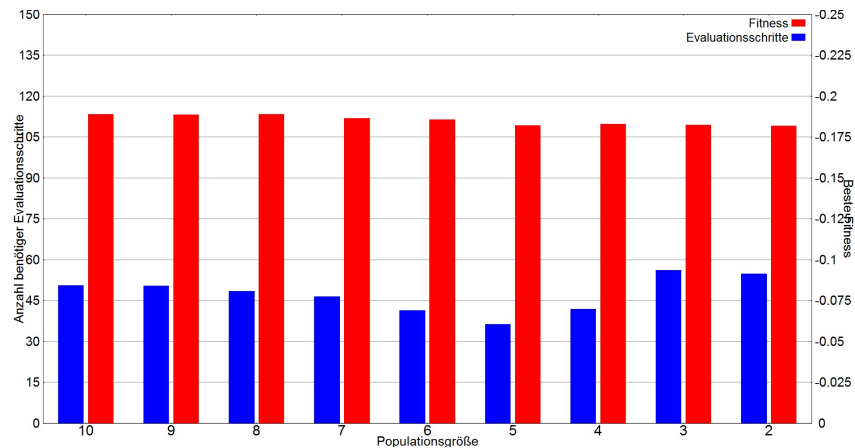


Abbildung 5.7: Geringste Anzahl benötigter Evaluationsschritte und beste gefundene Fitness für Populationen der Größen 2 bis 10

eine bessere Lösung zu finden also bei geringer Populationsgröße, da sich der Schwarm in der Regel weiter über den Suchraum ausbreitet. Die Abweichungen bzw. Verbesserungen zwischen den einzelnen Populationsgrößen sind aber nicht signifikant ausgeprägt. Die Partikel erreichen bei der betrachteten Zielfunktion immer die Randgebiete des Suchraums, sodass eine höhere Populationsgröße für mehr verarbeitete Stellen um Suchraum an den Randgebieten sorgt, aber nicht grundlegend signifikante Verbesserungen sorgt.

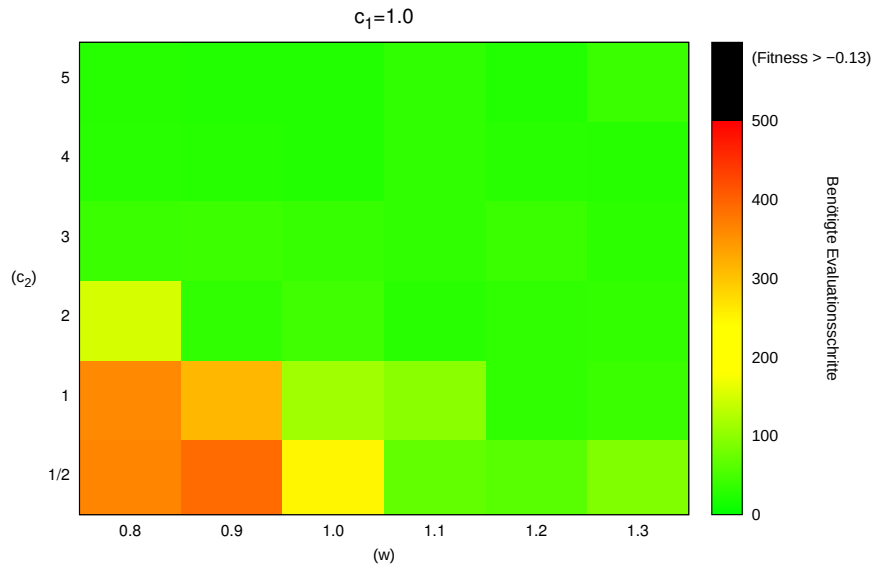


Abbildung 5.8: Gitterübersicht zur benötigten Anzahl an Evaluationschritten zur Grenzüberschreitung mit hohen Faktorstufen und Populationsgröße 5

5.4 Vergleich

Im vorherigen Abschnitt zeigen sich Beziehungen der Parameter w , c_1 , c_2 und S zur Effizienz des SPSO-11. Aufgrund der oben beschriebenen Trends, der Irrelevanz von c_1 und der ermittelten optimalen Populationsgröße werden die Parameter für einen ausführlicheren Vergleich mit den o.g. parameterfreien Algorithmen (*Zufallssuche*, *Hill Climber*, *Hooke-Algorithmus*) auf $w = 1.3$, $c_1 = 1$, $c_2 = 5$ und $S = 5$ gesetzt.

Alle Algorithmen durchlaufen 100 Durchgänge mit jeweils 200 Evaluationsschritten. Anhand dieser Daten werden für jeden Evaluationsschritt jeweils der schlechteste und beste Fitnesswert sowie der Durchschnitt und dessen Standardabweichung ermittelt, was in den Abbildungen 5.9 und 5.10 zu sehen ist. Daraus können Aussagen über jeweiligen Konvergenzen und Lösungsspektren der Algorithmen gemacht werden. Der SPSO-11 mit der neu ermittelten Parametrisierung ist um ein vielfaches besser als mit der Standardparametrisierung. Nach 25 Evaluationsschritten wird hier die ursprüngliche Grenze von -0.13 überschritten. Die zweite Variante des SPSO-11 Algorithmus benötigt so im Mittel nur noch $\frac{25}{274} \Rightarrow 9.12\%$ der Funktionsevaluationen der Variante mit Standardparametrisierung. Im Verhältnis zu den drei parameterfreien Algorithmen schneidet der SPSO-11 mit der ermittelten Parametrisierung recht gut ab. Der SPSO-11 ist gegenüber

der *Zufallssuche* und dem *Hill Climber* in allen Aspekten besser, sodass der entsprechende ernstzunehmende Konkurrent der *Hooke-Algorithmus* ist.

Der SPSO-11 hat zum Ende des Verlaufs den besten Wert was die insgesamt schlechteste Fitness in allen Evaluationsschritten angeht. Das verspricht eine gewisse Mindestqualität der gefundenen Lösungen, die besser ist als die beim *Hooke-Algorithmus*.

Was die insgesamt beste Fitness angeht, hat der *Hooke-Algorithmus* immer mal nach einigen Evaluationsschritten eine bessere Lösung gefunden, was beim SPSO-11 nicht der Fall ist. Nach 75 Evaluationsschritten hat der SPSO-11 die letzte beste Lösung mit einer Fitness von -0.243 gefunden, doch der *Hooke-Algorithmus* kommt sogar bis zu einer Fitness von -0.346, wenn auch erst nach dem 188. Evaluationsschritt. Die vom SPSO-11 gefundene beste Lösung wird vom *Hooke-Algorithmus* schon nach dem 38. Evaluationsschritt überboten.

Bei der Betrachtung des Mittelwertes fällt auf, dass der *Hooke-Algorithmus* und der SPSO-11 im Schnitt gegen den gleichen Fitnesswert konvergieren (≈ -0.19). Doch durch eine Eigenschaft unterscheiden sich die beiden Algorithmen von einander: Die Standardabweichung zum Mittelwert. Beim *Hooke-Algorithmus* steigt die Standardabweichung mit jedem Evaluationsschritt immer weiter an, was beim SPSO-11 nicht der Fall ist. Der SPSO-11 hat konstant eine Standardabweichung von ca. 0.02 und der *Hooke-Algorithmus* startet mit einer Standardabweichung von ungefähr 0.01 und geht im 200. Evaluationsschritt auf ungefähr 0.04 hoch. Im Verlauf der 200 Evaluationsschritte kann gesagt werden, dass die Standardabweichung der *Hooke-Algorithmus* linear wächst. Dies lässt sich auch schon anhand der vorigen Betrachtung der Verläufe der besten und schlechtesten Fitness vermuten. Da ist ersichtlich, dass das Spektrum der Fitnesswerte mit steigenden Evaluationsschritten des *Hooke-Algorithmus* immer größer werden. Beim SPSO-11 sind die Verläufe relativ flach, sodass sich die Standardabweichung nicht sonderlich sehr ändert.

In anbetracht der ermittelten Verläufe schneidet der SPSO-11 in der Tat am Besten ab, wenn auch nicht signifikant. Der SPSO-11 ist definitiv besser als die *Zufallssuche* oder der *Hill Climber*. Der *Hooke-Algorithmus* findet vereinzelt bessere Lösungen als der SPSO-11, aber durch die immer steigende Standardabweichung macht es den *Hooke-Algorithmus* unberechenbarer, sodass der SPSO-11 mit der konstanten Standardabweichung einen vorhersehbaren Verlauf zeigt.

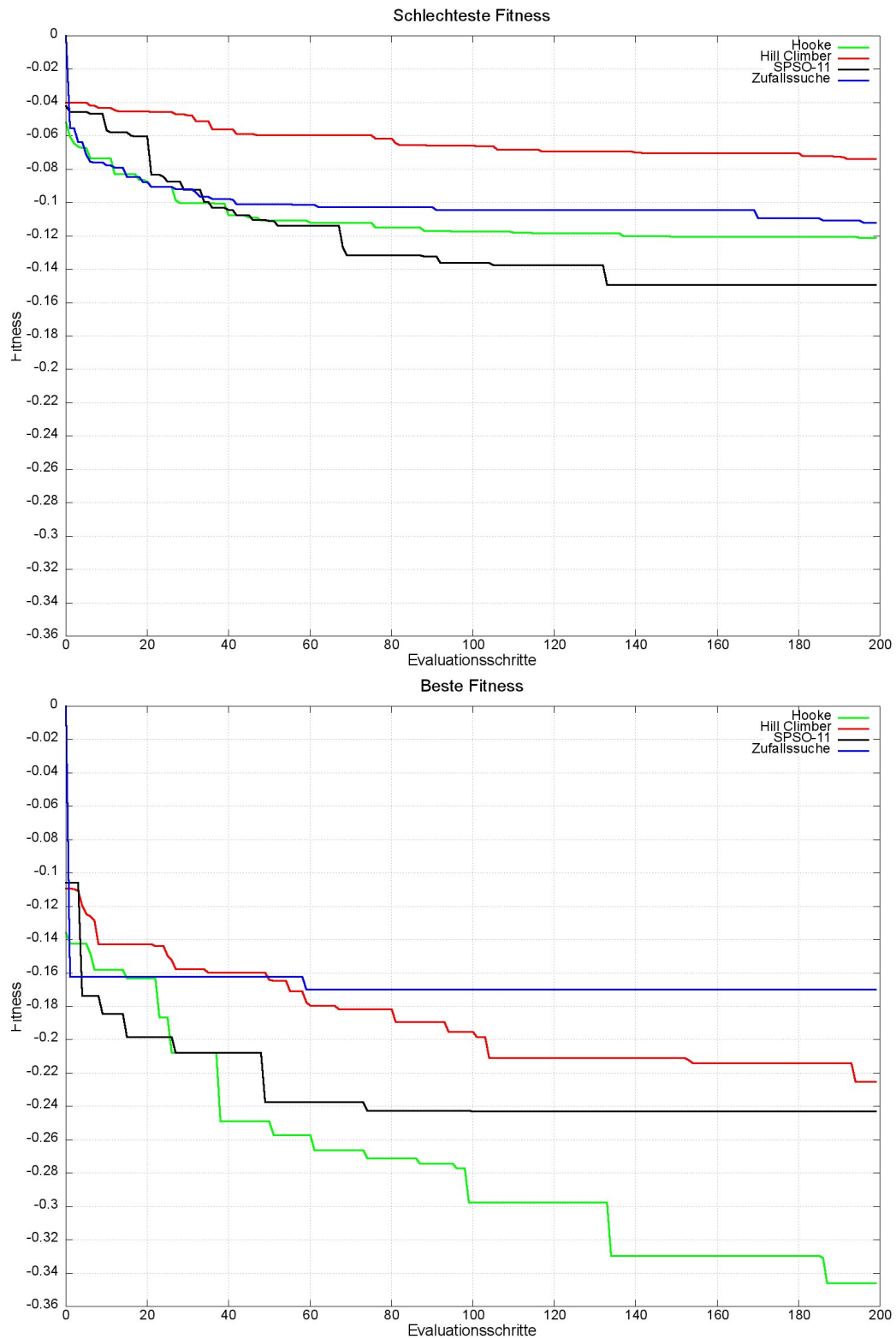


Abbildung 5.9: Schlechteste und beste Fitnesswerte der parameterfreien Algorithmen und dem SPSO-11 mit ermittelter Parametrisierung

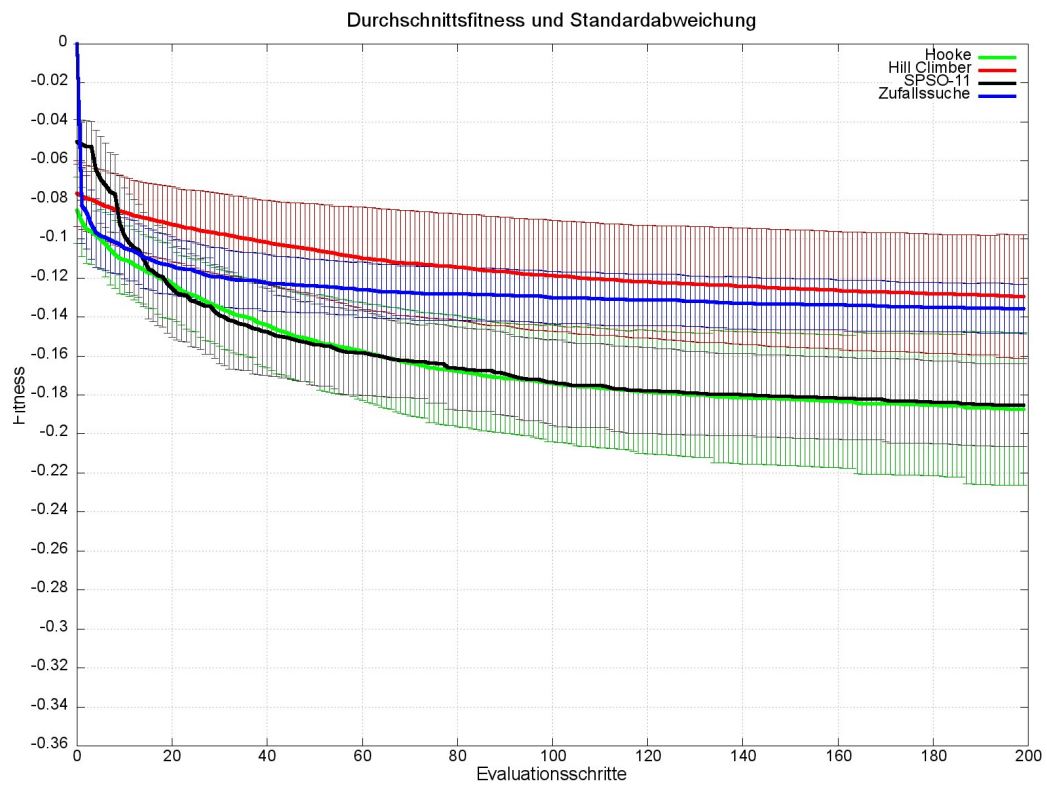


Abbildung 5.10: Durchschnitt und Standardabweichung von Fitnesswerten der parameterfreien Algorithmen und dem SPSO-11 mit ermittelter Parametrisierung

6 Schlussfolgerung

Es stellte sich heraus, dass die Zielfunktion bzw. das Optimierungsproblem zur Verringerung des Positionsfehlers des Endeffektors vom 3RUS nicht analytisch formulierbar ist, sodass auf ableitungsfreie Methoden zurückgegriffen wurde um Eigenschaften der Funktion experimentell zu ermitteln.

In der experimentellen Untersuchung wurde herausgestellt, dass die Zielfunktion nicht in Teilsuchräume separierbar ist, sodass die Komplexität der Funktion erhalten bleibt. Durch die Methodik der *Fitness Distance Correlation* wurde ermittelt, dass die Zielfunktion eine multimodale Funktion ohne Korrelation ist. Mit Hilfe der *Fitness Clouds* wurde entdeckt, dass die Steigungen innerhalb des Suchraums der Zielfunktion im Durchschnitt überschaubar sind. Desweiteren wurde anhand von Ausreißern in der *Fitness Cloud* herausgefunden, dass sich die besten Lösungen innerhalb des Suchraums an den Randgebieten des gesamten Suchraums befinden. Darauf hin konnte eine Reduzierung des Suchraums bestimmt werden.

Anhand dieser ermittelten Eigenschaften und dem neuen Suchraum stellte sich die Frage, ob der *Standard Particle Swarm Optimization 2011* für die Maßsynthese am 3RUS Parallelroboter zur Genauigkeitsteigerung anhand der Beurteilung am *Global Condition Index* geeignet ist. Um diese Frage zu klären wurde vorerst eine Parameteranalyse durchgeführt um eine geeignete Parametrisierung für den *Standard Particle Swarm Optimization 2011* zu finden.

Im Abschluss wurden die parameterfreien Algorithmen *Zufallssuche*, *Hill Climber* und *Hooke-Algorithmus* mit dem *Standard Particle Swarm Optimization 2011* mit ermittelter Parametrisierung verglichen. Unter allen vier Algorithmen ist der *Standard Particle Swarm Optimization 2011* für dieses Optimierungsproblem wenig überragend am Besten. Im Kontext zum Aufwand der experimentellen Untersuchung der Zielfunktion und Parameteranalyse des *Standard Particle Swarm Optimization 2011* ist die Entwicklung einer geeigneten Parametrisierung sehr aufwändig.

Falls der *Standard Particle Swarm Optimization 2011* für Maßsynthesen anderer Parallelroboter verwendet werden soll, ist dieser Aufwand unumgänglich um den *Standard Particle Swarm Optimization 2011* effizient zu gestalten. Auch nach den umfangreichen Untersuchungen steht das Ergebniss der Effizienz in keinem Verhältnis zum Aufwand der Ermittlung geeigneter Parametrisierungen. Da der *Hooke-Algorithmus* nur im Bezug der Standardabweichung gefundener Lösungen schlechter war, muss gesagt werden, dass der *Hooke-Algorithmus* Algorithmus insgesamt die Oberhand gewinnt, da keine grundlegenden Analysen aufgrund der Parameterfreiheit erforderlich sind.

Der *Standard Particle Swarm Optimization 2011* ist für die Maßsynthese zur Genauigkeitstei-

gerung am 3RUS Parallelroboter anhand der Beurteilung am *Global Condition Index* insgesamt im Vergleich zum Effizienzgewinn gegenüber des *Hooke-Algorithmus* zu aufwendig gewesen.

7 Ausblick

Durch das Vorwissen aus dem Kapitel der experimentellen Untersuchungen und der Betrachtung der gefundenen besten Lösungen innerhalb des Optimierungsproblems zeigte sich, dass diese Lösungen an den Randgebieten des Suchraum zu finden waren. Interessant zu ermitteln wäre, ob sich ein Algorithmus eignen würde der prinzipiell nur die Randgebiete in Augenschein nimmt. Diese Vorgehensweise würde den Suchraum des Optimierungsproblems um ein vielfaches verkleinern und könnte durch eine schneller ermittelte Lösungsfindung profitieren.

Desweiteren stellt sich die Frage, ob der *Global Condition Index* als alleiniger Gütewert für eine Stablängenkonfiguration ausreichend ist. In vielen Fällen innerhalb dieser Arbeit zeigte sich, dass Stablängenkonfigurationen in Schiefelage die Besten waren. Dies hat aber im Allgemeinen damit zu tun, dass ein kleiner Arbeitsraum entstanden ist. Dieser Arbeitsraum wird aufgrund weniger Auswertungen von Posen durch die numerische Annäherung des GCI in Gitterstruktur besser bewertet, da gerade die wenigen getroffenen Werte innerhalb des Arbeitsraums eine gute Fitness hervorbrachten. Der GCI ist eine gute Orientierung für die Güte einer Stablängenkonfiguration. Es sollte aber unbedingt der durch die Stablängenkonfiguration entstandene Arbeitsraum ins Verhältnis gesetzt werden um qualitativere Aussagen über die Stablängenkonfiguration zu machen.

Literatur

- [1] Sabbavarapu Ramana Babu, Vegesina Ramachandra Raju, and Koonam Ramji. Design for optimal performance of -RPS parallel manipulator using evolutionary algorithms. *Transactions of the Canadian Society for Mechanical Engineering*, 2013.
- [2] Manuel R. Barbosa, E.J. Solteiro Pires, and António M. Lopes. Optimization of Parallel Manipulators Using Evolutionary Algorithms. *SOCO 2010*, 2010.
- [3] Manuel R. Barbosa, E.J. Solteiro Pires, and António M. Lopes. Design of a Parallel Robotic Manipulator using Evolutionary Computing . *International Journal of Advanced Robotic Systems* 9, 2012.
- [4] Lukas Beyer. *Genauigkeitssteigerung von Industrierobotern*. Shaker Verlag, 2005.
- [5] Jason Brownlee. Benchmarking Algorithms. In *Clever Algorithms: Nature-Inspired Programming Recipes*. LuLu, 2012.
- [6] Philippe Cardou, Samuel Bouchard, and Clément Gosselin. Kinematic-Sensitivity Indices for Dimensionally Nonhomogeneous Jacobian Matrices. *IEEE TRANSACTIONS ON ROBOTICS*, VOL. 26, 2010.
- [7] Maurice Clerc. *Particle Swarm Optimization*. Iste Publishing Company, 2005.
- [8] Maurice Clerc. Standard Particle Swarm Optimisation. *Open access archive HAL*, 2012.
- [9] Sven Hensel. Modellierung und Optimierung von Werkzeugmaschinen mit parallelkinematischen Strukturen. Technical report, Verlag Wissenschaftliche Scripten, 2008.
- [10] Terry Jones and Stephanie Forrest. Fitness Distance Correlation as a Measure of Problem Difficulty for Genetic Algorithms. *Proceedings of the Sixth International Conference on Genetic Algorithms*, 1995.
- [11] Jens Kotlarski, Bodo Heimann, and Tobias Ortmaier. Influence of kinematic redundancy on the singularity-free workspace of parallel kinematic machines. In *Frontiers of Mechanical Engineering*. Higher Education Press, 2012.
- [12] Mathias Krefft. *Aufgabenangepasste Optimierung von Parallelstrukturen für Maschinen in der Produktionstechnik*. Vulkan Verlag, 2006.

- [13] António M. Lopes, Hélio Freire, P.B de Moura Oliveira, and E.J. Solteiro Pires. MaxiMin MOPSO Design of Parallel Robotic Manipulators. In *Soft Computing Models in Industrial and Environmental Applications, 6th International Conference SOCO 2011*. Springer Verlag, 2011.
- [14] António M. Lopes, Hélio Freire, P.B de Moura Oliveira, and E.J. Solteiro Pires. Multi-objective Optimization of a Parallel Robot using Particle Swarm Algorithm. *ENOC 2011*, 2011.
- [15] Jean-Pierre Merlet. Jacobian, manipulability, condition number and accuracy of parallel robots. *Robotics Research: Results of the 12th International Symposium ISRR*, 2007.
- [16] Collard Philippe, Verel Sébastien, and Clergue Manuel. Local search heuristics: Fitness Clouds versus Fitness Landscape. Technical report, University of Nice-Sophia Antipolis, 2007.
- [17] Andreas Pott. *Analyse und Synthese von Werkzeugmaschinen mit paralleler Kinematik*. Vulkan Verlag, 2006.
- [18] ABB Robotics. IRB 360 FlexPicker Data Sheet, 2013.
- [19] Marcel De Vegt. Einfluss verschiedener Parametrisierungen auf die Dynamik des Partikel-Schwarm-Verfahrens: Eine empirische Analyse. Technical report, Universität Dortmund, 2005.
- [20] Mauricio Zambrano-Bigiarinit, Maurice Clerc, and Rodrigo Rojas. Standard Particle Swarm Optimisation 2011 at CEC-2013: A baseline for future PSO improvements. *IEEE Congress on Evolutionary Computation*, 2013.

Abbildungsverzeichnis

2.1	Kin. Kette RUS	4
2.2	Skizze des 3RUS mit markierten Stäben	5
2.3	<i>Flexpicker</i> von <i>ABB Automatics</i> als annähernden Vergleich zum 3RUS	5
3.1	Veranschaulichung einer uniformen und quasi-zufälligen Verteilung	10
4.1	Vollständiger Permutationsbaum zur Separierbarkeitsüberprüfung	15
4.2	Reduzierter Permutationsbaum zur Separierbarkeitsüberprüfung	16
4.3	Plot der Fitness Distance Correlation	21
4.4	Plot der Fitness Cloud	24
4.5	Übersicht der zu untersuchenden Teilsuchräume zur Analyse des geeigneten Suchraums	26
4.6	Fitness Clouds der untersuchten Teilsuchräume zur Analyse des geeigneten Suchraums	27
4.7	Fitness Clouds des aufgespalteten streuenden Teilsuchraums zur Analyse des geeigneten Suchraums	27
5.1	Beispielhafte Veranschaulichung des Verhaltens von Partikeln	28
5.2	Übersicht der gefundenen Lösungen der zu vergleichenden parameterfreien Algorithmen und dem SPSO-11 mit Standardparametrisierung	33
5.3	Gitterübersicht zur benötigten Anzahl an Evaluationschritten zur Grenzüberschreitung mit niedrigen Faktorstufen und Populationsgröße 40	34
5.4	Gitterübersicht zur benötigten Anzahl an Evaluationschritten zur Grenzüberschreitung mit hohen Faktorstufen und Populationsgröße 40	35
5.5	Gitterübersicht zu Qualität der Lösung nach maximaler Anzahl an Evaluationschritten mit niedrigen Faktorstufen und Populationsgröße 40	36
5.6	Geringste Anzahl benötigter Evaluationsschritte und beste gefundene Fitness für Populationen der Größen 5, 10, 15, 20 und 40	38
5.7	Geringste Anzahl benötigter Evaluationsschritte und beste gefundene Fitness für Populationen der Größen 2 bis 10	38
5.8	Gitterübersicht zur benötigten Anzahl an Evaluationschritten zur Grenzüberschreitung mit hohen Faktorstufen und Populationsgröße 5	39
5.9	Schlechteste und beste Fitnesswerte der parameterfreien Algorithmen und dem SPSO-11 mit ermittelter Parametrisierung	41
5.10	Durchschnitt und Standardabweichung von Fitnesswerten der parameterfreien Algorithmen und dem SPSO-11 mit ermittelter Parametrisierung	42

Tabellenverzeichnis

4.1	Laufzeiten der Zielfunktion	18
4.2	Exemplarische Ergebnisse des modifizierten Hill-Climber Algorithmus	20
4.3	Exemplarische Parameter und Fitnesswerte der Ausreißer in der Fitness Cloud .	25
5.1	Übersicht der zu untersuchenden Faktorstufen zur Evaluation des SPSO-11 . .	32