

BACHELORTHESIS
Torben Schrader

Erarbeitung und Untersuchung eines Hybriden aus PSO und SPO.

FAKULTÄT TECHNIK UND INFORMATIK
Department Informatik

Faculty of Computer Science and Engineering
Department Computer Science

Torben Schrader

Erarbeitung und Untersuchung eines Hybriden aus PSO und SPO.

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung
im Studiengang *Bachelor of Science Angewandte Informatik*
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Michael Köhler-Bußmeier
Zweitgutachter: Prof. Dr. Michael Neitzke

Eingereicht am: 27. August 2020

Torben Schrader

Thema der Arbeit

Erarbeitung und Untersuchung eines Hybriden aus PSO und SPO.

Stichworte

PSO, Partikelschwarmoptimierung, SPO, Spiraloptimierung, Hybridisierung, Heterogene Partikelschwärme, Metaheuristik, Optimierungsverfahren

Kurzzusammenfassung

In dieser Arbeit wird ein erarbeiteter Hybrid aus Particle Swarm Optimization (PSO) und Spiral optimization (SPO) betrachtet und überprüft, ob er vergleichbare oder bessere Eigenschaften hat als reine PSO oder SPO Algorithmen. Es zeigt sich, dass mit nur geringen Anpassungen bessere Ergebnisse erzielt werden können als die einzelnen Implementierungen, allerdings die Anpassung teils recht spezifisch für eine Evaluationsfunktion sein kann und bei anderen Funktionen zu schlechteren Ergebnissen führen. Eine weiterführende Folgeuntersuchung wäre angebracht, besonders zur Untersuchung, ob gewisse Variationen allgemein bessere Ergebnisse erzielen, und um zu bestimmen, ob es spezielle SPO oder PSO Varianten gibt, welche besonders gut zusammenarbeiten.

Torben Schrader

Title of Thesis

Construction and examination of a hybrid of PSO and SPO

Keywords

PSO, Particle swarm optimization, SPO, Spiral optimization, Heterogenous particle swarm optimization, Metaheuristics, Optimization algorithms

Abstract

In this thesis a hybrid of SPO and PSO was researched and examined. It is shown that the hybrid is capable of achieving better results by combining SPO and PSO variants, though this enhancement can be quite specific for a certain kind of evaluation function,

and therefore needs specific tweaking of parameters for each kind of evaluation function. Further studies are recommended in order to determine the feasibility of using this hybrid with different SPO and PSO variants.

Inhaltsverzeichnis

Abbildungsverzeichnis	viii
1 Einleitung	1
1.1 Basiswissen	2
1.1.1 Optimierungsverfahren	2
1.1.2 Heuristiken	3
1.1.3 Metaheuristiken	3
2 Partikelschwarm- und Spiraloptimierung	5
2.1 Allgemeines Konzept der Partikelschwarmoptimierung (PSO)	5
2.2 Eigenschaften von PSO	6
2.2.1 Anwendbarkeit	6
2.2.2 Laufzeit und Präzision	6
2.3 Eigenschaften der PSO Parameter	7
2.3.1 Topologie / Nachbarschaft	7
2.3.2 Bewegungsfunktion	8
2.3.3 Partikel Anzahl	10
2.4 Spiraloptimierung (SPO)	10
2.4.1 Konzept	10
2.4.2 Eigenschaften	10
3 Konzept der Hybridisierung	12
3.1 Ziele und Eigenschaften	12
3.1.1 Exploration	12
3.1.2 Exploitation	13
3.2 Hybridisierung	13
3.2.1 Interaktion von PSO und SPO	14
3.2.2 Ausschlusszonen	14
3.2.3 Extrema Identifikation	15

3.2.4	Einschränkung der PSO und SPO Verfahren	17
3.3	Zusammenfassung	18
4	Implementation der Hybridisierung	19
4.1	Auswahl PSO und SPO	19
4.2	Hybrid	20
4.2.1	Exploratives Verfahren	20
4.2.2	Exploitatives Verfahren	21
4.2.3	Indikatorüberprüfung	21
4.2.4	Ausschlusszonen	21
4.2.5	Wrapper der Evaluationsfunktion	22
4.3	Ablauf	22
4.3.1	Initialisierung	22
4.3.2	Zyklus	22
4.3.3	Abbruch	23
4.3.4	Pseudocode	23
5	Eigenschaften und Grenzwerte der Implementierung	25
5.1	Eigenschaften	25
5.1.1	Äquivalenz zu den unabhängigen Implementierungen	25
5.1.2	Laufzeit	25
5.1.3	Kritische Eigenschaften	27
5.2	Grenzwerte	28
5.2.1	Interaktionsvarianten	28
6	Experimente	30
6.1	Rahmenbedingungen	30
6.1.1	Datei Struktur	30
6.1.2	Algorithmen und Evaluierungsfunktionen	31
6.2	Äquivalenzuntersuchung	31
6.2.1	Algorithmen und Evaluatoren	31
6.3	Determinierung geeigneter SPSO und SPO Varianten	35
6.3.1	SPO Varianten	36
6.3.2	PSO Varianten	38
6.4	Hybrid Variation	44
7	Auswertung und Verbesserungen	48

8 Abschluss	50
8.1 Zusammenfassung	50
8.2 Fazit	50
8.3 Ausblick	51
Literaturverzeichnis	52
A Anhang	53
A.1 Versuche zu den SPO Varianten	53
A.1.1 SPO Variation mit 2-Extrema	53
Selbstständigkeitserklärung	56

Abbildungsverzeichnis

6.1	Laufzeit Äquivalenz Test	33
6.2	Ergebnisse des Äquivalenztests der Laufzeit	33
6.3	Test zur Werte/Verhaltens Äquivalenzbestimmung	35
6.4	Testreihe für SPO Variation mit einem Extremum	37
6.5	Vergleich der besten SPO Varianten aus der Testreihe 6.4	39
6.6	SPSO Varianten mit 10-40 Partikeln	41
6.7	Vergleich verschiedener SPSO Varianten auf ihre Wahrscheinlichkeit, am dichtesten am globalen Extremum zu sein	42
6.8	SPSO Varianten mit 40 Partikeln und verändertem gpara	43
6.9	Werte Verhalten von einfachen Hybriden und Referenzen	45
6.10	Hybrid Werte Verhalten bei SPO Initialisierung nach fester Zyklen Anzahl	46
6.11	Hybrid Werte Verhalten bei SPO Initialisierung nach fester Zyklen Anzahl	47
A.1	SPO Variationen mit Konvergenzfaktor 0.95 und 0.9 mit SquareEval Eva- luator und 2 Maxima	54
A.2	SPO Variationen mit Konvergenzfaktor 0.85 und 0.8 mit SquareEval Eva- luator und 2 Maxima	55

1 Einleitung

Das Ziel dieser Arbeit ist es, eine neue Möglichkeit der Hybridisierung von PSO und SPO zu erarbeiten und zu untersuchen. Normalerweise ist es essentiell für ein PSO oder SPO Verfahren, ein gutes Gleichgewicht zwischen Exploration und Exploitation zu finden. Wenn die Exploration überwiegt, haben die Ergebnisse zwar das Potenzial bessere Ergebnisse zu erzielen, allerdings geht dies mit längeren Laufzeiten einher. Eine höhere Exploitation führt zu einer sehr schnellen und exakten Bestimmung eines Extremums, allerdings erhöht sich dabei die Gefahr, dass dieses Extremum nur ein lokales ist. Homogene Partikel Schwärmen haben meistens einen fließenden Übergang von Exploration zu Exploitation, parallel zur Konvergenz. Eine Alternative bieten heterogene Partikelschwärme[4]. Bei diesen Schwärmen können einzelne Partikel sich unterschiedlich Verhalten. Dies ermöglicht es Subschwärme zu haben, welche unterschiedliche Aufgaben übernehmen, wie Exploration und Exploitation. Dies ändert allerdings nicht die grundsätzlichen Eigenschaften der Schwärme, sondern ermöglicht unterstützende Effekte durch deren Interaktion. Einer dieser Effekte ist die Frage des Ausschlusses. Wenn mehrere Schwärme auf der gleichen Funktion arbeiten, besteht das Risiko, dass sie die selben Gebiete mehrfach untersuchen. Um dies zu verhindern, müssen die Schwärme miteinander kommunizieren. Dies verlangt eine Anpassung der Verfahren aufeinander. Diese Arbeit entstand aus der Möglichkeit, dass in einem heterogenen Partikelschwarm die Exploration und Exploitation nicht nur durch das Verhalten eines einzelnen Schwarmes bestimmt wird, sondern verschiedene Subschwärme unterschiedliche Aufgaben übernehmen können und miteinander kooperieren[4]. In Multi Schwärme (heterogen oder homogen) kooperieren verschiedene Schwärme miteinander, wofür diese in der Regel aufeinander abgestimmt und modifiziert werden müssen, um voneinander profitieren zu können. Dies erschwert es grundsätzliche Änderungen der Subschwärme vorzunehmen. In dieser Arbeit wird durch das Einführen von Ausschlusszonen eine Hybridisierung ermöglicht, in welcher die genutzten Subschwärme keine direkte Interaktion miteinander haben. Dadurch müssen diese nicht angepasst werden, um eine Interaktion mit andern Schwärmen zu ermöglichen, und können problemlos und ohne großen Aufwand ausgetauscht werden. Die Entkopplung führt

dazu, dass ebenfalls Hybride aus komplett unterschiedlichen metaheuristischen Verfahren gebildet werden können. Metaheuristische Hybride sind bereits seit längerem bekannt und wurden des öfteren als einfachen Metaheuristiken überlegen eingestuft[7]. Die konkrete Implementierung des Hybriden in dieser Arbeit ist auf PSO und SPO fokussiert, allerdings ist sie Grundsätzlich in der Lage, beliebige Verfahren zu Hybridisieren. Die Nutzung von Ausschlusszonen wie in dieser Arbeit wurde vom Autor zum Zeitpunkt der Veröffentlichung noch in keinem anderen Hybridverfahren gefunden.

1.1 Basiswissen

1.1.1 Optimierungsverfahren

Optimierungsverfahren sind Verfahren, welche versuchen die optimale Lösung für ein Problem zu finden. Das Ziel ist es, die einzige beste Lösung zu finden, was sowohl ein globales Minimum oder Maximum sein kann. Optimierungsverfahren finden überall Anwendung und sind häufig auftauchende Begriffe in Unternehmen (Workflow Optimierung) und der Industrie (Produktionsketten- und Transportkettenoptimierung). Optimierungsverfahren können grob in zwei Kategorien eingeteilt werden:

- perfekte Optimierungsverfahren
- sub-optimale Optimierungsverfahren

Während bei perfekten Optimierungsverfahren stets das Ziel ist, das eine beste Ergebnis zu finden und zu liefern, ist es bei sub-optimalen Verfahren das Ziel ein möglichst gutes Ergebnis zu liefern, auch ohne sicher zu sein, dass es kein besseres Ergebnis gibt. Während im Allgemeinen zwar die perfekte Lösung immer bevorzugt sein sollte, erfreuen sich sub-optimale Verfahren in vielen Bereichen großer Beliebtheit. Einer der Hauptgründe dafür ist die höhere Flexibilität von Annäherungsverfahren. Um perfekte Lösungen zu finden, müssen alle oder mindestens die relevanten Parameter des Problems bekannt und verstanden sein. Nur dann lässt sich ein Algorithmus erzeugen, welcher die perfekte Lösung findet. Eine Alternative wäre es die Probleme mit Brute-Force-Methoden zu lösen, wie dem systematischen Ausprobieren aller möglichen Lösungen und deren Vergleich. Diese Methode sollte zwar garantiert zur besten Lösung führen, hat allerdings mehrere gravierende Probleme. Zum einen müssen für die Garantie der optimalen Lösung alle Möglichkeiten ausprobiert werden. Dies ist zwar für kleinere Parametermengen

noch in annehmbarer Zeit realisierbar, allerdings gibt es Parameterräume, die so groß sind, dass es zwar theoretisch möglich wäre, alles auszuprobieren, allerdings praktisch mehrere Millionen Jahre oder länger brauchen würde. Des Weiteren gibt es unendliche Parameterräume bei denen ein vollständiges ausprobieren nicht möglich ist. Ist ein Problem allerdings gut genug bekannt, können sich selbst solche Probleme in annehmbarer Zeit perfekt lösen lassen, indem z.B. durch gefundene Metriken der Lösungsraum auf einen kleinen und endlichen Raum eingegrenzt werden kann. Dies ermöglicht zwar das Lösen von noch mehr Problemen, allerdings gibt es weiterhin gravierende Probleme. Eines sind Parameter im Bereich der reellen Zahlen. Ohne zusätzliche Kenntnisse, z.B über die Funktion an sich oder deren Eigenschaften, durch die sich mathematisch Lösungen errechnen lassen, ist es nahezu unmöglich perfekte Lösungen zu finden. Hier kommen meist sub-optimale Optimierungsverfahren zur Anwendung. Diese lassen sich abermals in zwei grobe Kategorien aufspalten:

- Heuristiken
- Metaheuristiken

1.1.2 Heuristiken

Bei Heuristiken handelt es sich um Annäherungsverfahren, welche anhand von Gesetzmäßigkeiten, Erfahrungen oder Wissen über die Eigenschaften des Lösungsraums versuchen eine Lösung zu finden. Dies bedeutet, dass sie durch Annahmen über den Lösungsraum versuchen, sich der besten Lösung zu nähern. Sofern diese Annahmen stimmen, sind Heuristiken auch sehr effektiv darin diese Probleme zu lösen, allerdings müssen diese korrekten Annahmen erst gefunden werden. Das Problem dieser Annahmen ist ebenfalls, dass sie nicht für alle Probleme gleich sind. Dies führt ebenfalls dazu, dass Heuristiken auf Grund ihrer Annahmen nicht in der Lage sind manche Probleme zu lösen. Dies ist unter anderem häufig der Fall bei nicht kontinuierlichen Parameterräumen.

1.1.3 Metaheuristiken

Während Heuristiken darauf fokussiert sind, anhand von Eigenschaften und Annahmen eine Lösung zu finden, lösen sich Metaheuristiken von diesen teils gänzlich ab. Diese Verfahren stützen sich häufig auf statistisch entdeckte Eigenschaften von Suchmuster. Diese Suchmuster sind zwar häufig für spezifische Arten von Funktionen am effektivsten,

allerdings können durch die komplette Loslösung von Annahmen und benötigtem Wissen, diese Metaheuristiken für nahezu alle Probleme genutzt werden. Dadurch, dass es sich bei ihnen nur um ein Näherungsverfahren handelt, haben sie zwar keine Gewähr die besten Lösungen zu finden, weisen im Allgemeinen dafür nur kurze Laufzeiten auf, was sie für eine Vielzahl von Problemen, besonders Echtzeitproblemen, sehr begehrt macht.

2 Partikelschwarm- und Spiraloptimierung

Nachdem im letzten Abschnitt ein allgemeiner Überblick über Optimierungsverfahren geschaffen wurde, wird in diesem Abschnitt näher auf verschiedene Varianten von Partikelschwarmoptimierung eingegangen und die Spiraloptimierung genauer betrachtet.

2.1 Allgemeines Konzept der Partikelschwarmoptimierung (PSO)

Partikelschwarmoptimierung ist ein Metaheuristisches Verfahren zur Bestimmung von einem Extremum einer Funktion, welches 1995 von R. Eberhardt und J. Kennedy[2] erstmals beschrieben wurde. Bei diesem Verfahren werden mehrere Partikel im Parameterraum initialisiert, welche Informationen enthalten über die Position des jeweiligen Partikels im Raum, den aktuellen Wert der Position, den höchsten Wert den der Partikel je bereits besucht hat und dessen Position, sowie eine Liste der Partikel, mit denen die Partikel ihre besten Ort / Werte Paare austauschen können (Nachbarschaft). Nach der Initialisierung wird über alle Partikel iteriert und mit Hilfe einer Bewegungsfunktion eine neue Position anhand der eigenen Position, der eigenen besten Position und der besten Position der Nachbarschaft berechnet und das Partikel auf diese Position verschoben. Dies wird so häufig wiederholt, bis entweder eine vorher bestimmte Anzahl an Iterationen durchlaufen wurde oder eine andere Abbruchbedingung erreicht wird. Nach der ersten Beschreibung 1995 wurden 1998 in der Arbeit von R. Eberhardt und Y. Shi[6] die Partikel um eine Trägheitsvariable erweitert, welche für ein Momentum bei der Berechnung der neuen Position sorgt. Dieser Parameter findet sich seitdem in fast allen Varianten von PSO wieder.

2.2 Eigenschaften von PSO

2.2.1 Anwendbarkeit

Wie aus der obigen Beschreibung zu erkennen ist, läuft der gesamte Algorithmus ohne Annahmen über den Funktionsgraphen ab, da nur die Werte bereits besuchter und aktueller Positionen genutzt werden. Dies bedeutet, dass die Struktur des Funktionsgraphen keinen Effekt auf den Ablauf von PSO hat, wodurch PSO auf allen Funktionen anwendbar ist, welche Werte liefern, bei denen eindeutig bestimmt werden kann, welcher Wert besser oder schlechter ist als ein anderer. Zudem ist es ebenfalls einfach, PSO Partikel für mehrdimensionale Eingangsparameter zu erweitern. Während in den meisten Fällen die Eingangsparameter und der Werteraum Teil der reellen Zahlen sind, ist es ebenfalls möglich die Bewegungsfunktion anzupassen, sodass sich die Partikel im Parameterraum aus ganzen Zahlen bewegen. Daraus folgt, dass sich PSO auf Funktionen anwenden lässt, welche als Parameter Objekte hat, sofern diese in einer passenden Matrix angeordnet werden können, über welche sich die Partikel bewegen können. Diese beiden Eigenschaften führen dazu, dass PSO für sehr viele Funktionen anwendbar ist, auch wenn nichts über den Funktionsgraphen bekannt ist, und fast immer ein Ergebnis liefern kann. Ein weiteres Indiz für die Fähigkeiten und Anwendbarkeit ist die Menge an unterschiedlichen Gebieten, in denen PSO genutzt wird [5].

2.2.2 Laufzeit und Präzision

Da es sich bei PSO um eine Metaheuristik handelt, bedeutet dies, dass der Algorithmus im Allgemeinen sehr schnell ist und in der Lage ist, Ergebnisse in unendlich großen Parameter- oder Lösungsräumen zu finden. Dies bedeutet allerdings ebenfalls, dass die Ergebnisse in der Regel nicht die exakten Extrema sind, sondern meist nur Annäherungen und es keine Gewähr dafür gibt, dass die Ergebnisse sich an globale oder lokale Extrema annähern. Aus diesem Grund eignet sich PSO hauptsächlich für Probleme, für die keine perfekten Ergebnisse nötig sind, und/oder diese Werte nicht in annehmbarer Zeit berechnet werden können. Ein weiterer Vorteil von PSO ist, dass es sich immer unterbrechen lässt und zu jedem Zeitpunkt ein Ergebnis liefern kann, wobei mit längerer Laufzeit die Wahrscheinlichkeit steigt, dass ein globales Extremum gefunden wird, und vor allem die

Genauigkeit der Bestimmung des gefundenen Extremums zunimmt. Somit ist PSO ebenfalls für Echtzeitanwendungen geeignet, da es eine Lösung bei einer begrenzten Laufzeit gewährleisten kann.

2.3 Eigenschaften der PSO Parameter

Die Hauptparameter von PSO, welche einen starken Einfluss auf das Verhalten des Verfahrens haben, sind:

- Nachbarschaft der Partikel (Topologie)
- Bewegungsfunktion und Trägheit
- Partikel Anzahl

2.3.1 Topologie / Nachbarschaft

Nachbarschaft der Partikel (Topologie)

Eine der Haupteigenschaften von PSO ist die Topologie der Nachbarschaft der Partikel, die bestimmt, welche Partikel ihre Werte mit welchen anderen Partikeln vergleichen können. Die Topologie kann sich sehr stark zwischen unterschiedlichen PSO Varianten unterscheiden und bestimmt zusammen mit der Bewegungsfunktion die Hauptmerkmale verschiedener Varianten.

Freie Topologie

Die Topologie kann die Eigenschaften in unterschiedlicher Weise verändern. So ist es beispielsweise möglich, die Topologie komplett frei zu lassen. Dies hätte zur Konsequenz, dass sich die Partikel untereinander nicht mehr austauschen und würde dazu führen, dass die Partikel sich eigenständig im Parameterraum bewegen, und eine relativ große Fläche abdecken würden. Allerdings geht dies mit einer nur sehr geringen Präzision einher, da, wenn ein Partikel ein Extremum findet, es alleine zu diesem konvergieren würde und die anderen Partikel sich ihm nur zufällig nähern würden. Des Weiteren besteht eine sehr große Gefahr, dass sich die Partikel in schlechten lokalen Extrema verfangen, da

sie zu schnell konvergieren um bessere Extrema zu finden. Um diesem entgegenzuwirken könnte man die Konvergenz der Partikel verringern und hierdurch eine höhere Abdeckung erreichen. Dies würde aber bedeuten, dass die Partikel sehr lange im Parameterraum umherschwirren müssen, was dann zu einer größeren Abdeckung und Laufzeit führen kann, bis man verlässlich präzise Extrema erhält und dadurch einen der starken Punkte von PSO zunichtemachen würde.

Statische Topologie

Viele PSO Varianten haben eine statische Topologie, was bedeutet, dass die Nachbarschaft der Partikel sich während der Laufzeit nicht verändert. Die Topologie kann von Variante zu Variante sehr unterschiedlich sein, wobei manche Varianten zufällig generierte oder ausgesuchte teilvernetzte Graphen oder auch voll vernetzte Graphen für ihre Topologie nutzen, ohne eine Topologie fest vorzugeben. Im Allgemeinen kann man davon ausgehen, dass umso stärker vernetzt die Topologie ist, desto schneller konvergieren die Partikel, da immer mehr Partikel zu den selben Punkten wollen, und umso weniger die Topologie vernetzt ist, desto stärker fällt die Exploration aus, da mehr kleine Gruppen entstehen, welche zu unterschiedlichen Orten wollen. Der Fokus auf Exploration oder Konvergenz/Exploitation hat je seine eigenen Vor- und Nachteile, worauf im folgenden Kapitel näher eingegangen wird.

Dynamische Topologie

Die dynamische Topologie ist ein Ansatz, um sowohl die Konvergenz von stark vernetzten Graphen und als auch die Exploration der leicht vernetzten Graphen zu nutzen. Hierbei wird während der Laufzeit die Topologie kontinuierlich geändert und ermöglicht es somit z.B. durch eine sehr lose Topologie am Anfang die Wahrscheinlichkeit zu erhöhen, das globale Extremum zu finden, und im Anschluss durch Erhöhen der Vernetzung die Konvergenz voranzutreiben, um schneller präzise Ergebnisse zu bekommen.

2.3.2 Bewegungsfunktion

Während die Topologie das durch den Informationsaustausch erzeugte Verhalten regelt, wird von der Bewegungsfunktion das Verhalten der einzelnen Partikel geregelt. Im allgemeinen ist die Bewegungsfunktion so gewählt, dass sich die Partikel mit jedem Schritt

dem ihnen bekannten stärksten Extremum annähern. Durch die Wahl, wie die bekannten Extrema und die eigene Position genutzt werden, kann die Konvergenz oder die Exploration erhöht werden. Eine Möglichkeit für die Bewegungsfunktion wäre, in jedem Schritt die Entfernung zum besten bekannten Extremum mit einem Faktor kleiner eins zu multiplizieren. Dies würde für jede Partikel Verteilung dazu führen, dass sie stark, deterministisch und linear konvergiert, allerdings kaum den Parameterraum erkundet und sehr stark von den initialen Positionen abhängt. Auf diese Weise ein gutes Ergebnis zu erreichen ist relativ unwahrscheinlich, da dieses Verfahren dazu neigt, sich in lokalen Extrema zu verfangen. Um die deterministischen Effekte zu verringern, wird meist eine zufällige Position in Abhängigkeit von der eigenen und der besten bekannten Position gewählt. Dies kann wieder linear sein, was hauptsächlich konvergentem Verhalten entspricht, oder z.B. radial, sodass ein zufälliger Punkt auf einem Kreis gewählt wird, dessen Mittelpunkt in der Mitte zwischen der eigenen Position und der besten bekannten Position liegt und dessen Durchmesser kleiner als der Abstand der beiden Punkte ist. Dies würde weiterhin die Konvergenz garantieren, resultiert aber in einem deutlich zufälligerem und vor allem explorativerem Verhalten, welches die Wahrscheinlichkeit stark erhöht, bessere Positionen zu finden. Ein weiterer Ansatz, um die Exploration zu stärken, wäre nicht nur die beste bekannte und eigene Position zu verwenden, sondern ebenfalls die beste selbst gefundene Position mit einzubeziehen. Auf diese Weise wird es ermöglicht, dass die Partikel nicht nur zum besten Punkt ihrer Nachbarschaft laufen, sondern mit einer gewissen Wahrscheinlichkeit auch ihre selbst entdeckten Extrema weiter erforschen. Ein ebenfalls oft genutzter Ansatz ist, eine Trägheit einzusetzen. Dies bedeutet, dass die Partikel auch ihre Bewegung von ihrer letzten Position in Betracht ziehen. Dies kann sowohl die Konvergenz als auch die Exploration unterstützen. Durch die Trägheit können Partikel über ihre eigentlichen Ziele hinausschießen und so neue Gebiete erkunden, die sehr weit von bekannten besten Orten entfernt sind, oder sich ihnen schneller nähern. Hierdurch kommt es zu einem oszillierenden Verhalten, sodass die Partikel immer wieder um und durch die Zielpositionen hindurch schreiten, wodurch sie die Umgebung dieser Punkte gleichmäßiger untersuchen und so verlässlichere und bessere Ergebnisse liefern können. Eine Möglichkeit eine wiederkehrende Exploration zu erzeugen, ist eine periodische Expansion. Dies bedeutet, dass die Partikel entgegen der Konvergenz getrieben werden, um wieder ein größeres Gebiet zu untersuchen und mögliche übersehene Extrema zu finden. Dies eignet sich besonders, wenn man den Algorithmus kontinuierlich arbeiten lassen will, da hierdurch eine Art Reset der Partikel erzeugt wird, sobald z.B. eine gewisse Konvergenz erreicht wurde oder genügend Zyklen durchlaufen wurden.

2.3.3 Partikel Anzahl

Die Partikel Anzahl hat keinen direkten Effekt auf einzelne Partikel, sondern wirkt sich auf die Topologie und besonders auf das Verhalten über die Evaluationen aus. Die Konvergenz von PSO hängt stärker vom Zyklus ab, bei dem alle Partikel einmal bewegt werden, als von der direkten Laufzeit. Das heißt ein PSO mit 10 Partikeln würde in etwa zehnmal so schnell konvergieren wie ein PSO mit 100 Partikeln wenn man die Evaluationen betrachtet, weil bei der selben Anzahl an Evaluationen die zehnfache Anzahl an Zyklen durchlaufen werden. Allerdings würde über die Zyklen gesehen der PSO mit 100 Partikeln, eine zehn mal höhere Menge an Positionen abgetastet haben und somit eine größere Abdeckung des Parameterraums erstellen. Die Betrachtung über die Zyklen wird besonders dann bedeutend, wenn PSO parallele Partikel Verarbeitung unterstützt.

2.4 Spiraloptimierung (SPO)

Spiraloptimierung ist ein relativ neues metaheuristisches Verfahren, welches erst 2011 von K. Tamura und K. Yasuda beschrieben wurde. Während es auf den ersten Blick einer starren Version von PSO ähnelt, handelt es sich doch um ein durchaus unterschiedliches Verfahren.

2.4.1 Konzept

Bei SPO wird ein Cluster von Abtastpunkten um ein Zentrum erstellt. Diese Abtastpunkte sind relativ zum Zentrum und drehen in jedem Schritt spiralförmig tiefer zum Zentrum. Dies passiert in Zyklen über alle Abtastpunkte, und nach jedem Zyklus wird geprüft, ob ein Abtastpunkt einen besseren Wert hat, als das Zentrum. Wenn dies der Fall ist, wird das Zentrum auf diesen Punkt verschoben und der nächste Zyklus beginnt. Die Abtastpunkte werden relativ zu ihrem Zentrum definiert über ihre Entfernung und Winkel, alternativ können sie auch über einen Vektor beschrieben werden.

2.4.2 Eigenschaften

Im Gegensatz zu PSO müssen die Abtastpunkte bei SPO weder ihren eigenen Wert merken, noch mit anderen Punkten Informationen auszutauschen. Dies resultiert in einem

sehr minimalistischen Algorithmus, der sowohl schnell, simpel aber trotzdem anpassbar und vielseitig ist. SPO nimmt wie PSO keine Annahmen über den Lösungsraum an. Prinzipiell ist SPO in der Lage auf die selben Funktionen wie PSO angewandt zu werden, mit Ausnahme von Funktionen mit ganzzahligen Parameterräumen, da für diese auf Grund der Neuartigkeit noch keine Varianten beschrieben wurden. Diese sind aber durchaus denkbar.

3 Konzept der Hybridisierung

Nachdem ein Überblick über die verschiedenen PSO und SPO Algorithmen geschaffen wurde, folgt nun die Erarbeitung des Hybriden. Zunächst wird festgelegt, welche Ziele und Eigenschaften erreicht werden sollen und im Anschluss dargelegt, welche PSO und SPO Verfahren mir am geeignetsten erscheinen, um diese zu erreichen und auf welche Weise die Hybridisierung stattfinden soll. Im folgenden Kapitel wird diese dann näher untersucht.

3.1 Ziele und Eigenschaften

Das Ziel dieser Arbeit ist die Erarbeitung eines neuen metaheuristischen Optimierungsverfahrens, welches in kurzer Zeit ein möglichst gutes Ergebnis liefert, mit zunehmender Laufzeit immer präziser wird, in der Lage ist, mehrere mögliche Lösungen zu liefern und bei dem mit längerer Laufzeit die Wahrscheinlichkeit steigt, das globale Maximum des Funktionsgraphen zu finden. Partikelschwarmoptimierungsverfahren sind in der Lage die meisten dieser Punkte zu erfüllen, allerdings muss ausbalanciert werden, inwiefern auf Exploration und Exploitation fokussiert wird und wie sich der Fokus während der Laufzeit ändert. In dieser Arbeit wird der Ansatz verfolgt, die Exploration und Exploitation voneinander zu trennen, in dem PSO genutzt wird um möglichst explorativ potenzielle Extrema Gebiete im Lösungsraum zu finden, während die Exploitation dieser Gebiete von einem SPO Verfahren übernommen wird, welches sehr stark auf Exploitation ausgelegt ist und parallel zum weiterlaufendem PSO die Extrema genau bestimmt.

3.1.1 Exploration

Die Fokussierung des PSO Verfahren auf die Exploration kann durch eine Verringerung der Konvergenz erreicht werden, was zur Folge hat, dass die Partikel länger aufgefächert den Lösungsraum durchwandern, bevor sie an einem Ort konvergieren und ihn genauer

abtasten. Während dies zwar die Laufzeit für präzise Ergebnisse verlängert, erhöht es die Wahrscheinlichkeit, dass noch unentdeckte Extrema gefunden werden. Die Idee bei der Hybridisierung ist, dass das PSO Verfahren kontinuierlich in einem stark aufgefächerten Zustand den Lösungsraum abtastet und wenn sich abzeichnet, dass ein potenzielles Extremum gefunden wurde, anstelle auf ein Konvergieren der Partikel zu warten, ein um das potenzielle Extremum konzentriertes SPO Verfahren eingeleitet wird. Während das SPO Verfahren nun versucht das Extremum zu bestimmen, sucht das PSO Verfahren parallel nach weiteren Extrema, und leitet bei entsprechenden Indizien ein weiteres SPO Verfahren ein.

3.1.2 Exploitation

SPO Verfahren ermöglichen es durch die Initialisierung der Abtastpunkte sehr dicht am Zentrum den explorativen Anteil sehr stark einzuschränken und direkt in einem stark exploitativen Zustand zu starten. Bei der Hybridisierung muss das SPO Verfahren nur relativ wenig explorieren, da dies bereits vom PSO Verfahren übernommen wurde. Dies bedeutet, dass wenn ein potenzielles Extremum gefunden wurde, ein SPO Zentrum an diesem Punkt gesetzt wird und die Abtastpunkte innerhalb der dazugehörigen Ausschlusszone initialisieren. Somit wird das potenzielle Extremum sofort stark ausgebeutet und nur ein sehr geringer Anteil des SPO Verfahrens wird genutzt, die Umgebung um ihn herum abzutasten.

3.2 Hybridisierung

Bei der Hybridisierung geht es um die Verknüpfung und der Steuerung von PSO und SPO. Dabei sollen die einzelnen Verfahren an sich nicht oder nur kaum von ihren eigentlichen Implementationen abweichen. Dies bedeutet ebenfalls, dass der Hybrid mit der richtigen Parameter Wahl in der Lage ist, das reine PSO oder SPO Verfahren zu simulieren. Bei der Hybridisierung müssen nun einige grundlegende Entscheidungen getroffen werden:

- Welche Interaktion findet zwischen PSO und SPO statt?
- Wie wird verhindert, dass immer nur die gleichen Extrema gefunden werden?
- Wie wird ein potenzielles Extremum festgestellt?

- Welche PSO und SPO Varianten müssen genutzt werden?

3.2.1 Interaktion von PSO und SPO

Es findet in dieser Hybridisierung keine direkte Interaktion zwischen dem PSO und dem SPO Verfahren statt. Die Hybridisierung kann als Gerüst verstanden werden, welches ein exploratives und ein exploitatives Verfahren einschließt, und diese beiden auf natürliche Weise steuert, basierend auf den Informationen, die es von beiden Verfahren erhält. Dies bedeutet, dass die Hybridisierung die kontinuierliche Exploration am Laufen hält, durch z.B. Neuinitialisierung und Zurücksetzen der Partikel, und die Exploitation für jedes gefundene Extremum initialisiert und gegebenenfalls terminiert wird. Es findet jedoch eine indirekte Interaktion zwischen den Verfahren über die Beeinflussung der Evaluierungsfunktion durch die Hybridisierung statt. Dies wird in folgendem Punkt genauer beschrieben.

3.2.2 Ausschlusszonen

Das Ziel der Hybridisierung ist es kontinuierlich weitere neue Extrema zu finden. Um dies zu erreichen müssen sowohl PSO als auch SPO über alle bisherigen gefundenen Extrema Bescheid wissen. Dies könnte erreicht werden, indem die PSO und SPO kontinuierlich Information über ihre Positionen austauschen, was eine starke Anpassung erfordern würde. Dies würde ebenfalls zu einer starken Verknüpfung der Verfahren führen und die Modularität stark beeinträchtigen. Eine Alternative dazu wäre das Einführen eines Wrappers für die Evaluierungsfunktion, welcher die Positionen der potenziellen Extrema kennt, und um diese herum Ausschlusszonen errichtet. Bei diesen Zonen gibt der Wrapper entweder ein Änti-Extremumßurück (ein Wert der immer schlechter als alle anderen angesehen wird) oder liefert einen Code, welcher von PSO und SPO erkannt wird, sodass sie die Evaluierung an diesem Punkt ignorieren. Auf diese Weise kann das explorative Verfahren die Bereiche um die potenziellen Extrema ignorieren und findet nur noch neue Extrema. Gleichzeitig kann man mit dem Code das SPO Verfahren innerhalb einer Ausschlusszone festhalten, sodass dieses nicht zu einem anderen Punkt wandert. Dies verursacht allerdings potenzielle Probleme. Es könnte dazu führen, dass sobald ein potenzielles Extremum gefunden worden ist, ein Gebiet um dieses herum nicht mehr vom explorativen Verfahren untersucht wird und das exploitative Verfahren nur ein Extremum

erkennt. Ebenfalls würde es dazu führen, dass der gesamte Lösungsraum ab einem gewissen Punkt nur noch aus Ausschlusszonen besteht und keine neuen Extrema gefunden werden können. Um diesem entgegenzuwirken, könnte die Größe der Ausschlusszonen anhand der fortschreitender Zeit, der prozentualen Abdeckung des Parameterraums durch Ausschlusszonen, der Indizien aus der Auswertung, z.B. zu viele Auswertungen die nicht gewertet werden, oder der Anzahl der gefunden potenziellen Extrema verringert werden. Eine weitere Anpassung wäre die Zentrierung der Zonen auf die gefundenen Extrema des entsprechenden SPO Zentrum, sollten diese terminiert werden. Auf diese Weise minimiert sich der unnötig versperrte Raum, und das Explorative Verfahren hat die größte Chance übersehene oder dicht beieinander liegende Extrema eventuell noch zu finden.

Zusammenfassung Ausschlusszonen:

- Ausschlusszonen werden erzeugt durch Wrapper der Evaluationsfunktion.
- Ausschlusszonen werden um potenzielle Extrema gelegt.
- Exploratives Verfahren darf sie nicht betreten oder ignoriert Werte innerhalb dieser Zonen.
- Exploitives Verfahren ist immer gebunden an eine Ausschlusszonen und kann nur innerhalb dieser Werte lesen.
- Beim Terminieren des exploitativen Verfahrens wird die entsprechende Ausschlusszonen auf das gefundene Extrema des Verfahrens zentriert.

3.2.3 Extrema Identifikation

Die Erkennung potenzieller Extrema ist stark davon abhängig welches explorative Verfahren gewählt wurde. Im Falle von PSO gibt es mehrere Möglichkeiten. Zum einen wäre es möglich zu warten, bis die Partikel eine gewisse Konvergenz erreicht haben, was allerdings bedeutet, dass die exploitativen Eigenschaften verstärkt werden müssen, um möglichst schnell ein potenzielles Extremum zu finden und würde gegen die Idee laufen, PSO auf die Exploration zu fokussieren. Zum anderen könnte man die Änderung der Position oder des Wertes des besten bis jetzt gefundenen Punktes nutzen. Sollte sich hier nach n Evaluierungszyklen nichts oder nur sehr wenig ändern, kann dies als Indiz für ein Extremum dienen. Eine weitere Möglichkeit wäre es, eine feste Anzahl an Zyklen zu

nehmen und nach deren Durchlauf die beste bis dahin gefundenen Position zu nehmen. Bei den letzten beiden Varianten kann es bei steigender Abdeckung des Parameterraums durch Ausschlusszonen zu schlechteren Ergebnissen führen, da die Wahrscheinlichkeit steigt, dass die Partikel sich innerhalb der ergebnen Ausschlussfläche befinden. Alle drei Varianten haben ihre Vor- und Nachteile:

Konvergenz :

Pro:

- Genaues Indiz
- Sehr kleine Ausschlusszonen möglich

Contra:

- Potenziell lange Laufzeit
- vollständiges Zurücksetzen der Partikel nötig (PSO abhängig)
- Verringerung der explorativen Eigenschaften für schnellere Ergebnisse

Änderung der Position/des Wertes über die Zeit :

Pro:

- Relativ genaues Indiz
- potenziell schneller als Konvergenz
- relativ kleine Ausschlusszonen möglich
- vollständiges Zurücksetzen der Partikel nicht unbedingt nötig (PSO abhängig)
- Keine Verringerung der explorativen Eigenschaften nötig

Contra:

- potenziell langsamer als Konvergenz
- Kann bei flachen Funktionsgraphen oder durch Zufall schnell zu schlechten potenziellen Extrema führen
- potenziell schlechtere Ergebnisse mit wachsender Ausschlussfläche

Feste Zyklenzahl :

Pro:

- Gleichmäßige Laufzeit
- Garantie relevanter Ergebnisse in fester Zeit
- Keine Informationen von PSO werden benötigt
- Bessere Kontrolle des Ablaufes

Contra:

- potenziell sehr unterschiedliche Indizien
- schlechtere Ergebnisse mit wachsender Ausschlussfläche

Alle drei Varianten haben ihre Berechtigungen für unterschiedliche Anwendungsbereiche. Im Laufe dieser Arbeit werden hauptsächlich Tests mit der dritten Variante durchgeführt und verglichen, da diese die beste Kontrolle des Ablaufes der Verfahren ermöglicht. Es wird allerdings erwartet, dass die Variante, die über die Positionsänderung geht, die am ehesten für diese Hybridisierung geeignete Variante ist, da sie kein Zurücksetzen oder Konvergieren der PSO Partikel benötigt, somit die kontinuierliche Exploration sehr gut unterstützt, und einen geringeren Glücks Faktor hat als die Variante mit fester Zyklenzahl, auch wenn dies auf Kosten längerer Laufzeit gehen kann.

3.2.4 Einschränkung der PSO und SPO Verfahren

Die in dieser Arbeit konstruierte Hybridisierung benötigt an sich keine Änderungen am eigentlichen PSO oder SPO Verfahren, sondern hat nur Einfluss auf die Evaluierungsfunktion, und extrahiert Informationen aus den Verfahren. Dies bedeutet, dass es keine richtig Beschränkung auf einzelne PSO oder SPO Verfahren gibt. Es wäre sogar möglich die Rollen zu vertauschen oder komplett andere Optimierungsverfahren zu nutzen, sofern diese Indizien für Extrema liefern können. Dies bedeutet nicht, dass alle Optimierungsverfahren geeignet sind, da die Hybridisierung darauf ausgelegt ist, die Eigenschaft der Verfahren zu nutzen auf Exploration oder Exploitation fokussiert werden zu können und die Evaluierungsfunktion zu beeinflussen. Verfahren, welche auf einen kontinuierlichen Funktionsgraphen angewiesen sind, sind deshalb nicht ohne weiteres nutzbar. In dieser Arbeit wird die Hybridisierung nur anhand weniger PSO und SPO Verfahren untersucht,

da die richtige Wahl und Nutzung dieser für einen konkreten Anwendungsfall wichtig ist, dies allerdings nicht nötig ist um zu zeigen, ob es prinzipiell durch die Hybridisierung zu einem Mehrwert kommen kann.

3.3 Zusammenfassung

In diesem Kapitel wurden die Grundkonzepte der Hybridisierung dargestellt und näher erläutert. Es handelt sich bei der Hybridisierung um ein Verfahren, welches, anstelle bei einem Verfahren die Exploration und Exploitation miteinander abzuwiegen, zwei unterschiedliche Verfahren für Exploitation und Exploration nutzt. Die Verfahren sind unabhängig von einander, fast unverändert bezüglich ihrer eigentlichen Implementation und können prinzipiell mit anderen Optimierungsverfahren ohne große Anpassungen ausgetauscht werden. Des Weiteren ist der Hybrid darauf ausgelegt, dass er nicht nur ein, sondern mehrere Extrema finden kann, und anstelle von ewiger Konvergenz, mit fortschreitender Laufzeit kontinuierlich den Lösungsraum stark explorativ abtastet.

4 Implementation der Hybridisierung

Nach dem im letzten Kapitel das Konzept der Hybridisierung erläutert wurde, wird im Folgenden die konkrete Implementation vorgestellt.

4.1 Auswahl PSO und SPO

Wie bereits erwähnt ist der vorgeschlagene Hybrid nicht fest an spezifische Optimierungsverfahren gebunden. Diese sind im Allgemeinen darauf ausgelegt, Exploration und Exploitation gegeneinander abzuwiegen, um selbstständig die besten Resultate zu erzielen. Das Anpassen eines PSO Verfahrens zu einem hauptsächlich explorativen Verhalten kann bei komplexeren und moderneren Varianten zu Problemen führen, da diese nicht darauf ausgelegt sind, die Exploitation größtenteils aufzugeben. Das Ziel dieser Arbeit ist es allerdings nicht, die Eignung von PSO Varianten für die Hybridisierung zu testen, sondern die Hybridisierung an sich auf Machbarkeit und Nutzen zu überprüfen. Aus diesem Grund wurden simplere Varianten für die Implementation gewählt, um ungewollte Nebeneffekte zu vermeiden und die Verfahren von ihrer ursprünglichen Form anzupassen. Eine separate Untersuchung von weiteren PSO Varianten für die Eignung dieser Hybridisierung wäre denkbar, sollte sich andeuten, dass es einen Gewinn durch die Hybridisierung gibt.

Für die Implementation des explorativen Verfahrens wurde Standard PSO (SPSO) 2011 [1] gewählt. Es handelt sich hierbei um eine sehr reine Implementierung vom Grundkonzept von PSO. Dies macht sie simpel und gut an die benötigten Anforderungen anpassbar.

Für die Implementation von SPO wurde die ursprüngliche Variante[8] genommen und leicht angepasst. Da es sich um ein relativ neues und nicht verbreitetes Verfahren handelt, gibt es keine große Vielfalt an Varianten zum Auswählen, zumal es auf Grund der simplen Arbeitsweise nicht viel Raum für Variationen gibt. Es wurden sowohl die ursprüngliche

Variante mit zufälliger Punkte Verteilung implementiert, als auch Varianten mit fest definierter Punkteverteilung. Dies ist nötig, da die Punkte Verteilung ein essenzieller Teil davon ist, wie stark die Exploitation ist.

4.2 Hybrid

Das Hybridverfahren besteht aus 5 Grundbausteinen, welche im Folgenden genauer erklärt werden.

- Dem explorativen Verfahren (SPSO)
- Dem exploitativen Verfahren (SPO)
- Der Indikatorüberprüfung
- Der Ausschlusszonen
- Dem Wrapper der Evaluierungsfunktion

4.2.1 Exploratives Verfahren

Für das ausgewählte SPSO wurde eine eigenständige Instanz implementiert, die unabhängig vom Hybriden nutzbar ist. Kontrolliert wird sie durch einen `cycle()` und einen `reset()` Befehl. `cycle()` lässt SPSO einen Zyklus absolvieren, sodass alle Partikel einmal bewegt werden, während `reset()` alle Partikel auf einen neuen Ausgangszustand zurücksetzt. Es kann von außen auf alle Partikel zugegriffen werden, dies sind aber nur lesende Zugriffe und die Verfahren benutzen zur Berechnung der Partikelwerte den Wrapper der Evaluierungsfunktion. Diese Implementation ist nicht in der Lage kontinuierlich explorativ zu arbeiten, sondern wird nach gewisser Zeit konvergieren. Aus diesem Grund muss sie in regelmäßigen Abständen zurückgesetzt werden, um ihre explorativen Eigenschaften nicht zu verlieren.

4.2.2 Exploitives Verfahren

SPO wurde genau wie SPSO als eigenständige Instanz implementiert. Es gibt drei leicht unterschiedliche Varianten der Implementation. Eine wie in [8] mit zufälliger Punkte Verteilung und eine mit einer spiralförmigen Anordnung, bei der alle Punkte auf einer Spiralbahn angeordnet sind, sodass die Bahn sich einmal um 360° um das Zentrum dreht und Punkte gleichmäßig und linear steigend vom Zentrum entfernt sind. Die dritte Variante hat drei Punkte, welche denselben sehr geringen Abstand zum Zentrum haben und um je 120° gedreht sind. Zudem werden weitere Punkte wie bei der zweiten Variante verteilt. Die erste Variante ist als Kontrollgruppe gedacht, während die zweite und dritte Variante gewählt wurden, da sie garantieren, Partikel in einem Mindestabstand für die Konvergenz zu haben. Während des Ablaufs des Hybriden können mehrere Instanzen von SPO parallel laufen, welche je eine eigene ID haben. SPO hat nur eine begrenzte Anzahl an Zyklen, die es durchschreitet. Wird diese Zyklen Anzahl erreicht, wird die Instanz terminiert.

4.2.3 Indikatorüberprüfung

Die Indikatorüberprüfung prüft nach jedem Zyklus vom SPSO Verfahren, ob eine Indikation eines Extremums vorliegt. Wird ein Indikator erkannt, wird der SPSO zurückgesetzt und ein SPO Cluster initialisiert, mit dem Zentrum auf dem besten bekannten Punkt des SPSO Verfahrens vor seinem Reset. Es wurden 3 Optionen implementiert. Die erste Option überprüft, wie viele Zyklen SPSO bereits durchlaufen hat. Die zweite Option überprüft, seit wie vielen Zyklen in Folge die beste gefundene Position sich weniger als der festgelegte Prozentsatz des Parameterraums bewegt. Die dritte Option überprüft, welche die durchschnittliche Distanz der Partikel ist. Identifiziert die gewählte Option ein Indiz, wird der beste gefundene Punkt von SPSO ermittelt, an diesem eine SPO Instanz initiiert, und dieser Punkt in die Ausschlusszonen Liste eingetragen. Anschließend wird SPSO zurückgesetzt.

4.2.4 Ausschlusszonen

Die Ausschlusszonen sind implementiert als eine Liste aus Positionen. Diese Punkte werden generiert, wenn ein SPO Cluster initialisiert wird und auf dessen Zentrum gesetzt. Die ID eines SPO Cluster stimmt mit der Position des Ausschlusszonen Punktes in der

Ausschlusszonen Liste überein. Sollte eine SPO Instanz terminieren, wird der Ausschlusszonen Punkt der zu dieser Instanz gehört auf deren letzte beste Position verschoben.

4.2.5 Wrapper der Evaluationsfunktion

Der Wrapper wird anstelle der eigentlichen Evaluationsfunktion an SPSO und SPO übergeben. Er hat Einsicht in die Ausschlusszonen Liste und besitzt eine Variable, welche angibt, wie groß die Ausschlusszonen sind. Wird der Wrapper nach einem Wert der Evaluierungsfunktion gefragt, überprüft er Anhand dessen ID, ob er zu der entsprechenden Position Zugriff hat. SPSO erfährt nur die Werte, die nicht in einer Ausschlusszone enthalten sind, während SPO nur Werte erhält, welche in der ihm zugehörigen Ausschlusszone enthalten sind.

4.3 Ablauf

Im Folgenden wird der reguläre Durchlauf des Hybriden näher erläutert.

4.3.1 Initialisierung

Bei der Initialisierung werden dem Hybriden ein Dummy PSO Verfahren und ein Dummy SPO Verfahren, zusammen mit den grundlegenden Parameter zur Initialisierung, wie der Ausschlusszonen Breite, der maximalen Zyklen oder Evaluierungsanzahl, der Evaluierungsfunktion, und den Bedingungen für die SPO Initialisierung, sowie Sonderbedingungen für den Ablauf, wie dem verbieten des PSO Zurücksetzens, übergeben. In dieser Phase wird zudem anhand der PSO-Dummy Vorlage das PSO Verfahren initialisiert.

4.3.2 Zyklus

Der Zyklus des Hybriden kann in fünf Phasen aufgeteilt werden:

- 1. Der PSO Zyklus: Sofern nicht bereits an einer Abbruchbedingung angelangt, wird als erstes der PSO Zyklus durchlaufen

- 2. Die Initialisierungsüberprüfung: Nach dem PSO einen Zyklus abgeschlossen hat, wird geprüft, ob alle Bedingungen zur Bildung eines neuen SPO Clusters erfüllt sind.
- 3. Die SPO Initialisierung: Wird in Schritt 2 festgestellt, dass ein neuer SPO erzeugt werden muss, wird an der aktuell besten bekannten Position von PSO ein SPO Cluster gesetzt. Im Anschluss wird eine neue Ausschlusszone auf das Zentrum des Clusters gesetzt. Abschließend wird, sofern erlaubt, PSO auf einen Initialzustand zurückgesetzt.
- 4. SPO Zyklen: Es wird über alle vorhandenen SPO Cluster iteriert und jeder Cluster durchläuft einen Zyklus durchlaufen.
- 4.2 SPO Terminierung: Erreicht eine SPO ihre Abbruchbedingung, wird sie aus der Liste vorhandener SPO entfernt und ihre zugehörige Ausschlusszone wird auf die beste bekannte Position der SPO zentriert

4.3.3 Abbruch

] Wird die Abbruchbedingung des Hybriden während eines Zyklus erreicht, bricht das Verfahren ab und sperrt sich. Es kann so modifiziert werden, dass es direkt seine beste bekannte Position samt Wert zurückgibt, oder passiv auf Abfrage wartet. Es kann das insgesamt beste Position/Wert Paar zurückgegeben werden, oder eine Liste aller gefundenen Extrema. Diese Liste ergibt sich aus den besten Positionen der PSO und SPO Verfahren, sowie den Zentren der Ausschlusszonen bereits terminierter SPO Verfahren.

4.3.4 Pseudocode

Vereinfachte Darstellung des Verlaufes als Pseudocode:

```
SPO-list = [];  
exclusionzones = [];  
initialize SPSO;  
while endcondition not met do  
    SPSO.cycle();  
    if Extremum indicator detected then  
        SPO-list.append(new SPO at indicator.pos);  
        exclusionzones.append(indicator.pos);  
    end  
    foreach SPO in SPO-list do  
        SPO.cycle();  
        if SPO at maxCycle then  
            exclusionzones[SPO.ID] = SPO.center;  
            terminate SPO;  
        end  
    end  
end  
return best or all best points found by SPSO and SPO;
```

Algorithm 1: Ablauf des Hybridverfahrens

5 Eigenschaften und Grenzwerte der Implementierung

Nachdem in den letzten beiden Kapiteln Konzept und Implementation der Hybridisierung dargestellt wurden, betrachten wir nun ihre Eigenschaften und Grenzwerte näher.

5.1 Eigenschaften

5.1.1 Äquivalenz zu den unabhängigen Implementierungen

Wie erkennbar ist, unterscheiden sich die eigenständigen PSO und SPO Implementationen nicht von den im Hybriden eingebauten Algorithmen, da auf diese nur lesend zugegriffen und nicht ihre Verhaltensweise modifiziert wird. Dies bedeutet, dass sich der Hybrid, wenn er nur PSO oder SPO verwendet, äquivalent zu deren eigenständigen Implementationen verhält.

5.1.2 Laufzeit

Wenn der Hybrid nur SPO oder PSO nutzt, und im Werte Verhalten äquivalent ist, gibt es eine leichte Verschlechterung der Laufzeit durch den Overhead des Hybriden. Dieser bleibt konstant und erhöht somit die Laufzeit von jedem Hybrid Zyklus um einen festen Wert. Angenommen die Dauer eines Zyklus $T_{Alg,p}$ vom Algorithmus Alg mit p für die Anzahl der Evaluierungen pro Zyklus, bleibt unverändert bei steigender Zyklenzahl n , dann kann die Zyklus Laufzeit des Hybriden $T_{Hyb}(n)$ mit Hilfe der Formel 5.1 beschrieben werden. T_O steht hier für die Dauer des Overhead pro Zyklus.

$$LZ(n) = (T_O + T_{Alg}) \quad (5.1)$$

Somit steigt die Laufzeit des Grenzwert Hybriden bei steigender Zyklenzahl linear in Abhängigkeit zur Laufzeit des relevanten Algorithmus an. Die Prämisse, dass die Dauer eines Zyklus sich nicht ändert, gilt im Hybriden allerdings nur solange, wie entweder nur PSO oder SPO genutzt wird. Sobald beide zusammen genutzt werden, entstehen durch den Hybrid Algorithmus im Laufe der Zyklen Ausschlusszonen. Während SPO in der hier untersuchten Variante nur an eine Ausschlusszone gebunden ist, zieht PSO zur Evaluierung alle Ausschlusszonen in Betracht, um zu berechnen ob die Evaluierung sich innerhalb einer von ihnen befindet. Dies kann mit Hilfe der Formel 5.2 beschrieben werden. $T_{EXC}(n_{exc}(n))$ steht hier für die Dauer der Bestimmung, ob sich eine Evaluierung innerhalb einer Ausschlusszone befindet, in Abhängigkeit von der Anzahl an Ausschlusszonen. $n_{exc}(n)$ gibt die Anzahl an Ausschlusszonen zu jedem Zeitpunkt an. Dieser Wert ist sehr stark von der jeweiligen Implementation des Hybriden abhängig, hat allerdings immer die Eigenschaft mit zunehmender Zyklenzahl stetig oder bis zu einem Grenzwert zu steigen.

$$LZ(n) = (T_O + (T_{PSO,p1} + T_{Exc}(n_{exc}(n)) * p1) + (T_{SPO,p2} + T_{Exc}(1) * p2) * n_{SPO}(n)) \quad (5.2)$$

$n_{SPO}(n)$ gibt hier an, wie viele SPO Cluster in jedem Zyklus vorhanden sind. Die Anzahl der SPO Cluster ist abhängig von den Initialisierungsregeln und den Abbruchbedingungen:

- Stetig steigend, sofern immer mehr Initialisierungen als Abbrüche stattfinden.
- Begrenzt stetig steigend, sofern immer mehr Initialisierungen als Abbrüche stattfinden, es aber einer Art von Limit gibt
- Dynamisches Gleichgewicht, sofern Anzahl an Initialisierungen und Abbrüchen sich aufheben, Fluktuationen um ein Gleichgewicht sind möglich
- Einzel/Gruppen Cluster, sofern SPO Abbrechen bevor neu Initialisierungen stattfinden

Diese Auflistung schließt die allgemeinsten Fälle ein, mit der Wahl der Initialisierungsregeln und der Abbruchbedingungen lassen sich jedoch auch komplexere Verhalten bilden. In der Formel 5.2 ist erkennbar, dass die Laufzeit der Zyklen des Hybriden von 3 Komponenten abhängig ist:

- Einem konstanten Overhead T_O
- Der PSO Zyklus Laufzeit $T_{PSO,p1} + T_{Exc}(n_{exc}(n)) * p1$
- Der SPO Zyklus Laufzeit und Anzahl $(T_{SPO,p2} + T_{Exc}(1) * p2) * n_{SPO}(n)$

Während die erste Komponente konstant bleibt, ist die zweite Komponente zusätzlich mit der Zyklenzahl (begrenzt) stetig steigend und resultiert somit in einem über-linearen Anstieg der Laufzeit bei steigender Zyklenzahl. Die Anzahl der SPO Cluster der dritten Komponente kann zwar ebenfalls stetig steigend sein, allerdings ist davon auszugehen, dass es sich meist um ein dynamisches Gleichgewicht oder Einzel/Gruppen Cluster handeln wird. Diese Komponente kann somit im Schnitt ebenfalls als konstant betrachtet werden. Einen weiteren nicht zu vernachlässigenden Einfluss auf die Laufzeit hat die Evaluationsfunktion. Die Evaluationen sind vergleichbar mit der Taktrate aller in dieser Arbeit behandelten Algorithmen. Somit ist zu erwarten, dass bei langsameren Evaluatoren die Laufzeit merklich ansteigt.

5.1.3 Kritische Eigenschaften

Da im Hybriden PSO und SPO ohne direkte Einflussnahme ablaufen, übernimmt der Hybrid zu Teilen deren Eigenschaften. Dies bedeutet, dass die Wahl der passenden PSO und SPO Varianten von großer Bedeutung ist, wenn der Hybrid optimale Leistung erbringen soll. Eine andere Eigenschaft der Implementierung ist, dass mit fortlaufender Zeit nicht nur wie gewünscht immer mehr des Lösungsraumes durch Ausschlusszonen versperrt wird und somit PSO zur Exploration neuer Gebiete gezwungen wird, sondern auch nach einer gewissen Zeit der gesamte Parameterraum blockiert wird, was effektiv zum Stillstand des Hybriden führt. Dies ist allerdings hauptsächlich bei sehr langen Laufzeiten bei kleinen Ausschlusszonen, oder unpraktikabel großen Ausschlusszonen ein Problem. Während dieser Arbeit wird hierauf nicht tiefer eingegangen, allerdings wäre eine potenzielle Maßnahme wie bereits erwähnt, die Ausschlusszonen Größe mit der Laufzeit zu verringern. Somit würden nach und nach alte Gebiete freigelegt, welche aufgrund ihrer Lage am

Rand der Zone noch nicht stark untersucht worden sind, und die kontinuierliche Arbeit am Laufen hält.

5.2 Grenzwerte

5.2.1 Interaktionsvarianten

Wie bereits in der vorherigen Sektion angesprochen, ist der Hybrid in der Lage so angepasst zu werden, dass er verschiedene Verhalten annimmt. Im Folgenden werden einige Grenzfälle aufgelistet und genauer erläutert.

Hybrid_PSO

In dieser Variante findet keine Hybridisierung statt, sondern es wird nur PSO ausgeführt. Dies kann erreicht werden, indem die Initialisierungsbedingung für SPO so gewählt wird, dass SPO nie initialisiert wird. Der Hybrid verhält sich in diesem Fall identisch zur entsprechenden PSO.

Hybrid_SPO

In dieser Variante findet ebenfalls keine Hybridisierung statt, sondern wird PSO direkt übersprungen und direkt ein SPO Cluster initialisiert. Erreicht wird dieser Fall, indem beim Initialisieren des Hybrides der `disablePSO` Parameter gesetzt wird. Durch diese Einstellung wird PSO als terminiert markiert und sein zurücksetzen verhindert.

Einzeln sequenziell

Dieser Fall bedeutet, dass zuerst PSO bis zu einer Abbruchbedingung durchlaufen wird und im Anschluss zu SPO gewechselt wird. Der Hybrid verhält sich als ob erst PSO und anschließend auf dessen Ergebnis zentriert SPO ausgeführt wird. Hierzu reicht es den `noreset` Parameter auf `True` zusetzen. Auf diese Weise terminiert PSO sobald die Initialisierungsbedingung erfüllt ist und wird nicht zurückgesetzt.

Kontinuierlich sequentiell

Hierbei stehen PSO und SPO in einem dynamischen Verhältnis zueinander. Es wird mit PSO angefangen, bis die Initialisierungsbedingung erfüllt ist. Bei der Initialisierung des Clusters wird gleichzeitig die Ausschlusszone des Clusters gesetzt und PSO wird zurückgesetzt. Wenn sowohl PSO als auch SPO vorhanden sind, wird in jedem Zyklus einmal der PSO Zyklus und im Anschluss von jedem aktiven SPO Cluster ein Zyklus durchlaufen. Dies ist das Standardverhalten des Hybriden. Es wird zwischen Exploration und Exploitation gewechselt und das Verhältnis wird bestimmt durch die Anzahl der Partikel/Suchpunkte von PSO/SPO und der Anzahl der gleichzeitig aktiven SPO Cluster.

6 Experimente

Nach Darlegung und Bewertung der Implementierung der erarbeiteten Hybridisierung werden im Folgenden Experimente durchgeführt, um die Erwartungen zu überprüfen und ein Fazit zu ziehen, ob es sich lohnt, die Hybridisierung in Zukunft umfangreicher zu untersuchen.

6.1 Rahmenbedingungen

Alle folgenden Experimente wurden in Python umgesetzt und mit Hilfe von matplotlib visualisiert. Python wurde als Programmiersprache gewählt, da dem Autor die Sprache vertraut ist, sich schnell darin programmieren lässt. Mit matplotlib steht bereits eine einfache aber umfangreiche Möglichkeit zur Verfügung, die Ergebnisse zu visualisieren. Des Weiteren wurden die grundlegenden Strukturen sowie Teile des Codes aus der Klausurleistung/Hausarbeit des Autors vom Modul "Complex Adaptive Systems" letzten Semesters mit nur leichten Änderungen adaptiert. Die übernommenen Code Fragmente beinhalten hauptsächlich die Grundfunktionalität der SPSO und der Standard Evaluator Implementation. Die folgenden Experimente sind alle beschränkt auf den 2 Dimensionalen Parameterraum ($X_Achse = \{x | x \geq -10 \wedge x \leq 10 \wedge x \in \mathbb{R}\}$, $Y_Achse = \{y | y \geq -10 \wedge y \leq 10 \wedge y \in \mathbb{R}\}$) und es wird nur nach Maxima gesucht, da es dadurch einheitlicher ist und für das Verhalten der Algorithmen keinen Unterschied macht, ob sie nach dem Maximum oder Minimum suchen. Die Tests werden in Relation zu den Evaluierungen und nicht den Zyklen gemacht, da sich auf diese Weise Variationen mit unterschiedlichen Partikel Anzahlen direkt vergleichen lassen.

6.1.1 Datei Struktur

Die Arbeit ist auf vier Dateien aufgeteilt. In der AlgoLib.py sind die Implementatioen aller Algorithmen als eigenständige Klassen enthalten. Zusätzlich enthält diese Datei eine

Kopierfunktion, welche eine auf den Anfangszustand zurückgesetzte Kopie des übergebenen Algorithmus liefert. Die `EvalLib.py` enthält alle Evaluierungsfunktionen und sind ebenfalls als Klassen definiert. Die Evaluierungsfunktionen sind Unterklassen von `Evaluator` und überschreiben die Funktion `calc_val(pos)`, wobei `pos` ein array der x und y Koordinate des Punktes zum evaluieren ist. Die `TestCases.py` enthält alle Testfunktionen und Testabläufe. Alle Ergebnisse sind in `AllLogs.csv` geschrieben. Die `BachelorUtil.py` enthält eine Hilfsfunktion, um die Einträge in der `AllLogs.csv` schnell und einfach neu zu plotten. Der Anhang zur Arbeit befindet sich auf CD und kann beim Erstgutachter eingesehen werden.

6.1.2 Algorithmen und Evaluierungsfunktionen

Als PSO Algorithmus wurde SPSO 2011 implementiert, da es sich hierbei um eine simple Grundlage für PSO handelt. Da der Fokus dieser Arbeit auf der Untersuchung des Verhaltens des Hybriden und nicht der vollständigen Optimierung liegt, wurden noch keine anderen PSO Kandidaten implementiert, damit die Wechselwirkung der Eigenschaften von PSO mit dem Hybriden, ungestört von spezifischen Verhaltensweisen verschiedener PSO_Varianten, untersucht werden kann. Für den exploitativen Teil wurde die ursprüngliche SPO Version[8] mit einer leichten Änderungen in der Punkte Initialisierung implementiert.

6.2 Äquivalenzuntersuchung

Wie in Section 5.1.1 genauer erklärt, sollte der Hybrid in der Lage sein, sowohl PSO als auch SPO mit nur geringen Unterschieden in Laufzeit und Verhalten nachzubilden. Diese Erwartung wird im Folgenden untersucht.

6.2.1 Algorithmen und Evaluatoren

Zum Testen wurden fünf Algorithmen miteinander verglichen. Die Parameter und die Wahl der SPO Variante sind willkürlich, da für diesen Test die Werte der Ergebnisse unerheblich sind und nur der relative Vergleich von Interesse ist:

- SPSO Willkürliche Initialwerte mit 10 Partikeln

- SPO_Even_Spiral Willkürliche Initialwerte mit 10 Partikeln
- Hybrid-PSO Wie in 5.2 beschrieben initialisiert und als PSO und SPO werden die gerade erstellten SPSO und SPO_Even_Spiral genommen
- Hybrid-SPO Wie in 5.2 beschrieben initialisiert und als PSO und SPO werden die gerade erstellten SPSO und SPO_Even_Spiral genommen
- Hybrid Standard Variante des Hybriden. Für PSO und SPO werden die gerade erstellten SPSO und eine leichte Variation von SPO_Even_Spiral genommen. Die Initialisierungsbedingung eines neuen SPO ist das Warten von 30 Zyklen und die Abbruchbedingung für die SPO ist das Erreichen von 30 Zyklen. Diese Bedingungen sind ebenfalls willkürlich gesetzt, mit Werten die zu diesem Zeitpunkt sinnvoll erschienen.

Es wurden zwei Durchläufe durchgeführt, die mit Ausnahmen des Evaluators die selben Parameter haben. Als Evaluator wurden zwei LinearEval Evaluatoren genutzt, einer mit einem Extremum und der andere mit zehn Extrema. Die Algorithmen die den Evaluator mit zehn Extrema nutzen, sind gekennzeichnet durch ein "_Multiim Namen. Genauere Informationen zu den Parametern sind hier nicht von Relevanz, können jedoch bei Bedarf in der Log Datei des Anhangs nachgesehen werden.

Laufzeit Äquivalenz

In der Abbildung 6.1 ist zu erkennen, dass SPSO und Hybrid-PSO sowie SPO_Even_Spiral und Hybrid-SPO wie erwartet von der Laufzeit her nahezu identisch sind. In der vergrößerten Darstellung 6.2b lässt sich ein konstanter Unterschied zwischen den reinen Implementationen und den spezialisierten Hybriden feststellen. Dies entspricht dem erwarteten Overhead des Hybriden aus Absatz 5.1.2. Ebenfalls ist das überlineare Verhalten des normalen Hybriden erkennbar, welches bei höherer Laufzeit wie in 6.2a noch deutlicher ist. Beim Betrachten des Laufzeitverhaltens des Hybriden fällt auf, dass er anfänglich äquivalent mit SPSO ist, allerdings nach kurzer Zeit leicht schneller wird, nur um anschließend wieder kontinuierlich langsamer zu werden. Dieses Verhalten lässt sich dadurch erklären, dass der normale Hybrid bis zur ersten Initialisierung eines SPO Clusters identisch zum PSO spezialisierten Hybriden ist, und sich somit auch wie SPSO verhält. Die temporäre Abnahme lässt sich dadurch erklären, dass nun die SPO Cluster einen Teil der Evaluationen ausmachen. Da diese schneller sind als SPSO, nimmt hier

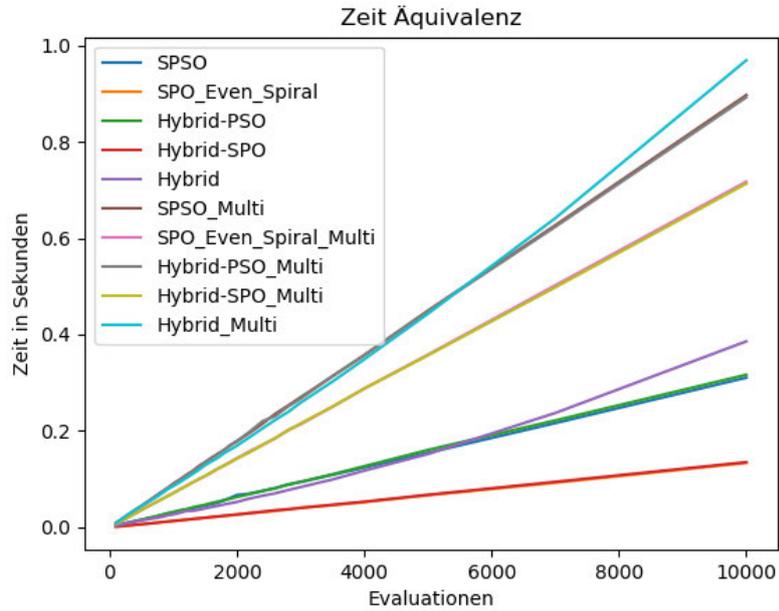
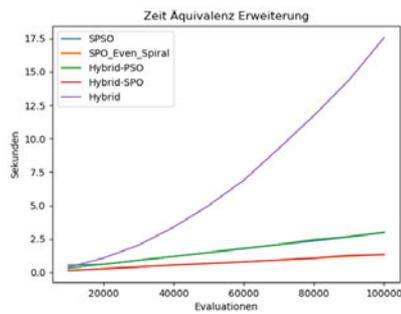
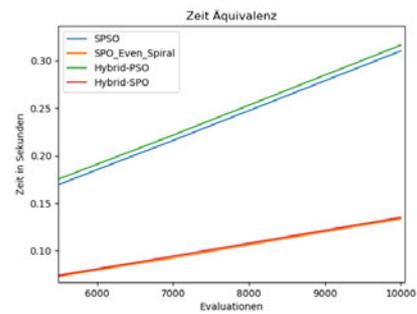


Abbildung 6.1: Laufzeit Äquivalenz Test



(a) Hybrid Laufzeit



(b) Overhead Darstellung

Abbildung 6.2: Ergebnisse des Äquivalenztests der Laufzeit

auch die Laufzeit des Hybriden ab. Da sowohl SPO als auch PSO lineare Laufzeiten haben, würde die Mischung im Hybriden in einer Laufzeit zwischen den beiden Verfahren resultieren, abhängig vom Verhältnis von SPO und PSO Evaluationen pro Hybrid Zyklus. Auf Grund des über-linearen Verhaltens fällt dies jedoch nur minimal aus und der Hybrid wird stetig langsamer. Dies Verhalten deckt sich mit den Erwartungen aus Absatz 5.1.2, ebenso wie der erwartete Einfluss der Evaluationsdauer. Die Dauer der Berechnung des LinearEval Evaluators steigt auf Grund der Implementierung linear mit der Extrema Anzahl an, was sich in den deutlich längeren Laufzeiten widerspiegelt. Allerdings ist ebenfalls zu erkennen, dass dies keinen Einfluss auf das Laufzeit Verhalten an sich hat und die einzelnen Verfahren die selben Abstände haben, wie bei der schnelleren 1-Extrema Variante.

Verhaltens/Werte Äquivalenz

Nachdem die Laufzeit Äquivalenz bereits bestätigt wurde, wird nun die Verhaltens Äquivalenz untersucht. Da es sich bei SPSO um ein teils zufälliges Verfahren handelt, ist es schwer das Verhalten direkt miteinander zu vergleichen. Stattdessen wurde das Werte Verhalten über die Evaluationsanzahl verglichen. Während die SPO Implementation zwar deterministisch ist, setzt die Initialisierung SPO auf eine zufällige Position, weswegen, ebenso wie bei SPSO das Werte Verhalten betrachtet wird. Es wurden dieselben Evaluatoren und Algorithmen genutzt wie während des Äquivalenztests.

Wie in 6.3 erkennbar ist, verhalten sich PSO und SPO werte-äquivalent zu ihren spezialisierten Hybrid Varianten. Ebenfalls ist wieder erkennbar, dass sich der Hybrid zu Beginn äquivalent zur spezialisierten PSO-Hybrid Variante verhält. Zudem zeigt sich bereits, dass der Hybrid in der Lage ist, ohne Veränderung der ursprünglichen Algorithmen bessere Ergebnisse zu erzielen als diese allein. Es ist zu erkennen, dass SPO und PSO zu Werten unter 100 konvergieren. Dies spiegelt weniger wider, dass die Verfahren nicht in der Lage sind, das globale Extremum von 100 genau zu bestimmen, sondern dass sie teilweise nur lokale Extrema finden. Dies bedeutet, dass das Ergebnis ein Indiz ist, wie verlässlich die Verfahren das globale Maximum finden. Hier zeigt sich, dass PSO und SPO zwischen 250 und 600 Evaluationen keine signifikanten Verbesserungen haben, was bedeutet, dass in diesem Bereich beide Verfahren von Exploration zu Exploitation wechseln und gegen Ende kein relevanter explorativer Anteil mehr vorhanden ist. Beim Hybriden ist jedoch erkennbar, dass er auch noch nach 1000 Evaluationen einen explorativen Anteil besitzt und er somit verlässlicher das globale Extremum findet.

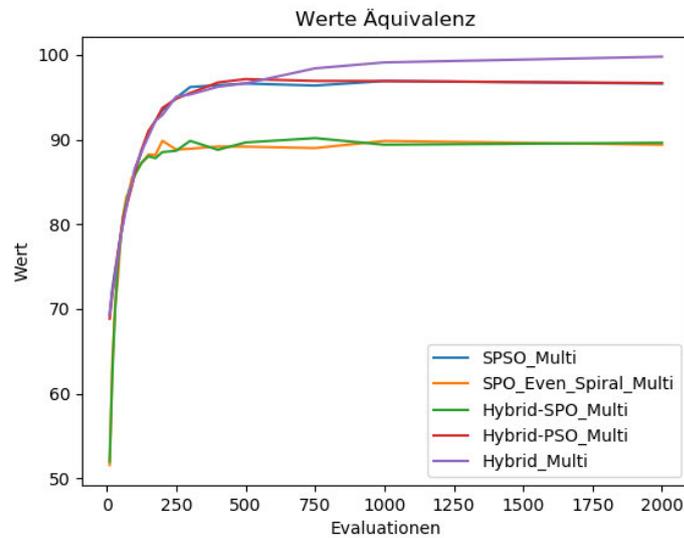


Abbildung 6.3: Test zur Werte/Verhaltens Äquivalenzbestimmung

Zusammenfassung Äquivalenz

Aus den vorherigen Tests ist zu erkennen, dass der Hybrid so konfiguriert werden kann, dass er nahezu identisch ist mit den in ihm integrierten PSO oder SPO Verfahren. Dies spiegelt die Eigenschaft wider, dass der Hybrid keinen direkten Einfluss auf das Verhaltens oder die Laufzeit der integrierten Algorithmen hat. Ebenfalls ist zu sehen, dass der Hybrid bei geringeren Evaluationen von der Laufzeit der integrierten Verfahren nur leicht abweicht und erst bei relativ vielen Evaluationen deutlich schlechter wird. Dies ist allerdings nicht sehr problematisch, da es sich hierbei um sehr schnelle Verfahren handelt, die jenseits von 2000-3000 Evaluationen häufig keine signifikant besseren Ergebnisse liefern. Des Weiteren deuten sich bereits die Eigenschaften der kontinuierlichen Exploration an, durch die der Hybrid sein durchschnittliches Ergebnis kontinuierlich zum globalen Extremum hin verbessern kann.

6.3 Determinierung geeigneter SPSO und SPO Varianten

Nach dem zuvor wahllose Parameter gewählt wurden und sich im Hybriden bereits positive Effekte gezeigt haben, wird nun experimentell untersucht, wie SPSO und SPO

angepasst werden können, um den gewünschten explorativen und exploitativen Eigenschaften gerecht zu werden.

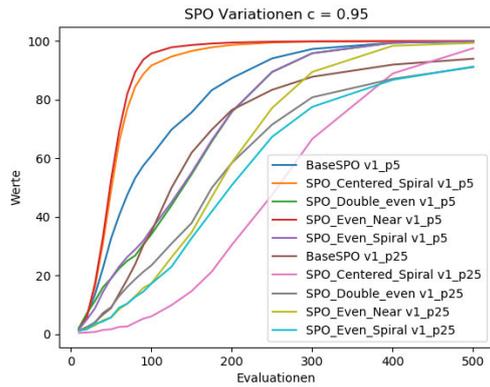
6.3.1 SPO Varianten

Das Ziel der SPO Variationen ist, das Verfahren so zu modifizieren, dass es nur noch einen minimalen explorativen Anteil hat und sehr stark konvergiert. Dies wird zwar die Wahrscheinlichkeit stark erhöhen, sich in einem lokalen Extremum zu verfangen, im Ideal Fall ist dies aber nicht von Relevanz, da es Aufgabe von PSO ist, ein einzelnes Extremum zu lokalisieren und SPO nur noch dieses untersuchen muss. Somit ist das Ziel, eine SPO Variation zu finden, die für 1-Extremum Probleme die schnellsten und besten Ergebnisse liefert. Es wurden vier SPO Varianten implementiert. Diese Varianten verhalten sich identisch, unterscheiden sich aber in ihrer Punkte Verteilung. Während BaseSPO eine zufällige Konfiguration hat, ähnlich wie die ursprüngliche Implementation[8], haben die anderen Varianten fest definierte Verteilungen:

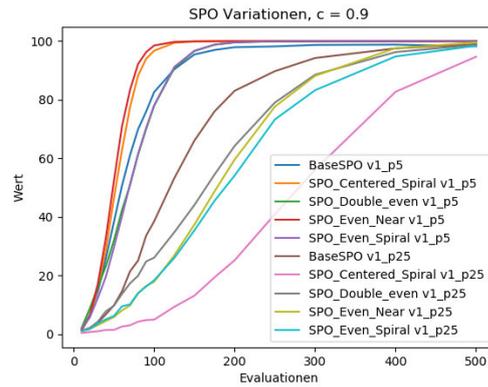
- SPO_Even_Spiral: Die Punkte entfernen sich mit linear wachsenden Abständen vom Zentrum und sind um $360^\circ/\text{Anzahl der Punkte}$ zum nächsten Punkt versetzt, sodass sich eine gleichmäßige Spirale bildet
- SPO_Even_Near: Identisch zu SPO_Even_Spiral, allerdings sind drei der Partikel extra dicht ums Zentrum gelegt, um die Exploitation zu fördern.
- SPO_Centered_Spiral: Wieder eine Spirale, die sich um 360° dreht, nur nimmt hier der Abstand zum Zentrum quadratisch ab, wodurch eine deutlich höhere Punkte Konzentration am Zentrum vorliegt.
- SPO_Double_near: Gleiche Implementation wie SPO_Even_Spiral, nur dass sich die Spirale um 720° Grad dreht statt um 360°

Die folgenden Experimente nutzen als Evaluator einen SquareEval mit einem einzelnen Extremum der Größe 100. Die anzupassenden Parameter sind Partikel Anzahl(p), die Partikel Konfiguration und die Konvergenz Variable(c).

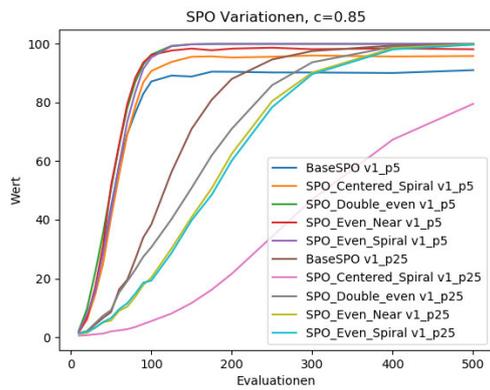
In Abbildungsreihe 6.4 ist klar zu erkennen, dass im Allgemeinen SPO Cluster mit weniger Partikeln schneller konvergieren als die mit mehr Partikeln. Dies liegt daran, dass die Cluster mit jedem Zyklus kleiner werden, und somit bei kleineren Clustern weniger



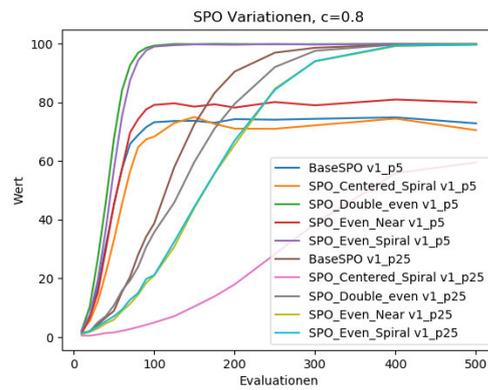
(a) SPO Variationen $c=0.95$



(b) SPO Variationen $c=0.9$



(c) SPO Variationen $c=0.85$



(d) SPO Variationen $c=0.8$

Abbildung 6.4: Testreihe für SPO Variation mit einem Extremum

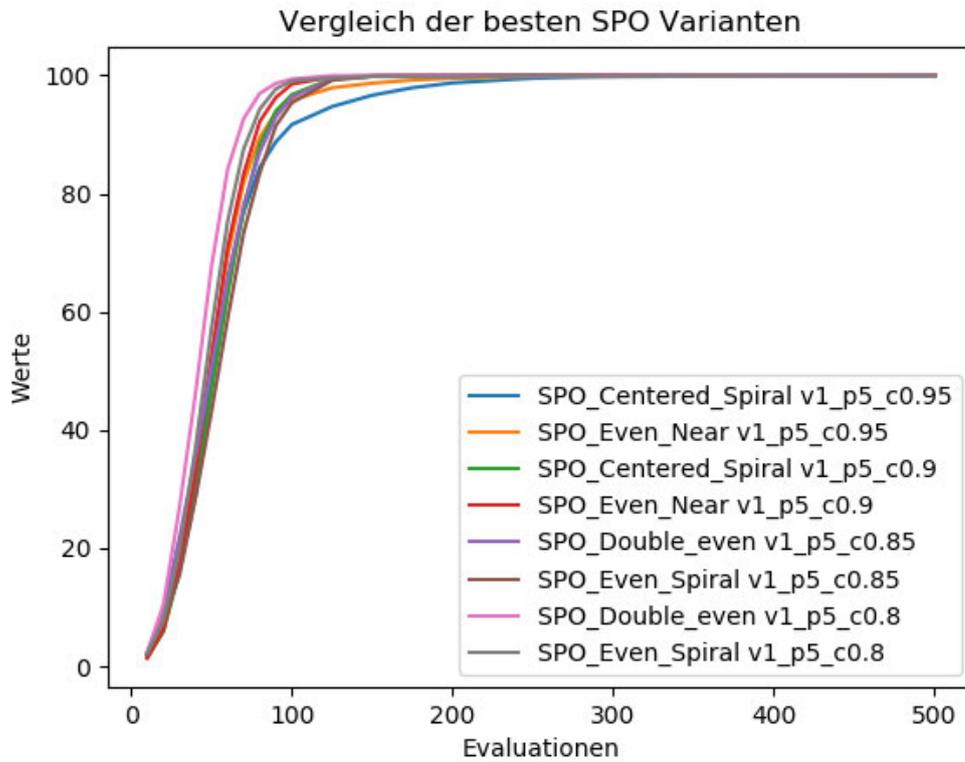
Evaluationen zu einem Zyklus gehören. In dieser Reihe ist ebenfalls erkennbar, bestimmte Varianten für unterschiedliche Konvergenzen besser geeignet sind. Für die Auswahl, welche SPO Variante für den Hybriden am geeignetsten ist, wurden die besten Varianten zusammen in Abbildung 6.5 dargestellt und genauer betrachtet.

Während in 6.5a SPO_Double_Even ($p=5, c=0.8$) am besten zu sein scheint, ist in der vergrößerten Ansicht 6.5b zu sehen, dass diese Variante zwar am schnellsten konvergiert, sie aber keine sehr hohe Präzision oder Verlässlichkeit hat, was an dem springenden Verlauf erkennbar ist. Dasselbe gilt für SPO_Even_Spiral ($p=5, c=0.8$), nur dass hier das Springen früher und stärker auftritt, wodurch es sich nicht lohnt sich diese weiter zu betrachten. Als sehr gute Kandidaten zeigen sich SPO_Even_Near ($p=5, c=0.9$), SPO_Double_Even ($p=5, c=0.85$) und SPO_Even_Spiral ($p=5, c=0.85$), wobei die beiden letzteren Varianten nur sehr geringe Unterschiede aufweisen, mit sehr leicht besseren Ergebnissen für SPO_Double_Even. Während der erste Kandidat eine anfangs deutlich stärkere Konvergenz aufweist, wird bei genauerer Betrachtung deutlich, dass die späteren beiden Kandidaten ab ca. 200 Evaluationen dauerhaft bessere Werte liefern, diese sich aber ab diesem Punkt nur noch sehr geringfügig unterscheiden. Die anderen Variationen werden im Folgenden nicht weiter betrachtet, da sie zu jedem Zeitpunkt schlechter als eine der gerade genannten Variationen sind. Zur Auswahl für spätere Hybrid Variationen bieten sich drei Variationen an:

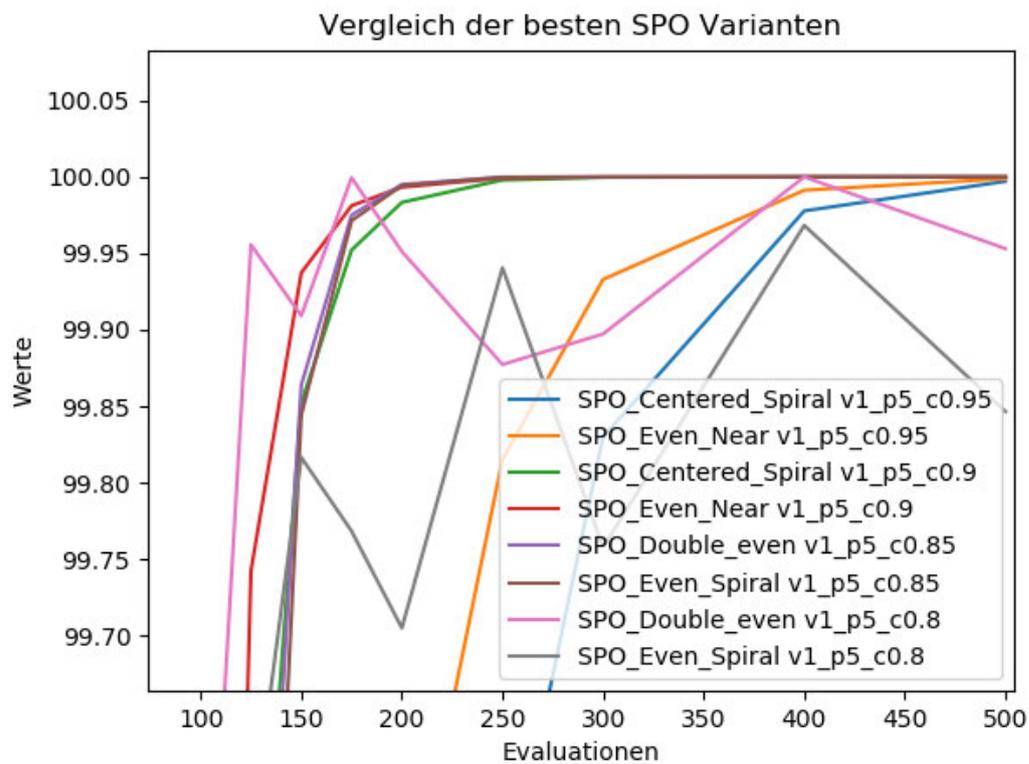
- Wenige Evaluation (<125): SPO_Double_Even ($p=5, c=0.8$)
- Mittel viele Evaluation (125-200): SPO_Even_Near ($p=5, c=0.9$)
- Viele Evaluationen (>200): SPO_Double_Even ($p=5, c=0.85$)

6.3.2 PSO Varianten

Das gewünschte Ergebnis der PSO Variationen ist, dass das Verfahren möglichst lange und stark explorativ verläuft und anstelle eine präzisen Bestimmung eines Extremums, eine Position liefert, die mit einer möglichst hohen Wahrscheinlichkeit dichter am globalen Extremum ist, als an einem lokalen. In dieser Arbeit wird sich nur mit Standard PSO[1] mit einer Ring Topologie beschäftigt. Diese Entscheidung wurde getroffen, da eine genaue Optimierung eines PSO Algorithmus auf die gewünschten Eigenschaften den zeitlichen Rahmen sprengen würde, zumal sich gezeigt hat, dass es keine allgemeine optimale Topologie für alle Funktionen gibt[3]. Ebenfalls hat es sich gezeigt, dass lose



(a) Vergleich der besten SPO Varianten



(b) Vergrößerung auf 99.7-100

Abbildung 6.5: Vergleich der besten SPO Varianten aus der Testreihe 6.4

Topologien explorativeres Verhalten hervorrufen, als stark vernetzte[3], weshalb für diese initiale Untersuchung eine einfache Ring Topologie gewählt wurde. Des Weiteren ist das Ziel dieser Arbeit nicht das Finden des besten explorativen PSO Verfahrens, sondern die initiale Bewertung, ob diese Form der Hybridisierung prinzipiell in der Lage ist, nützlichere Ergebnisse zu liefern als die explorativen und exploitativen Verfahren einzeln, ohne diese groß zu verändern. Bei der Untersuchung von SPSO werden die Partikel Anzahl, die Größe der Nachbarschaft der Ring Topologie und der rFactor variiert. Der rFactor bestimmt, wie stark die Partikel konvergieren und ist eine leichte Anpassung der originalen SPSO Implementierung. Ein Wert von 1 ist equivalent zum Original, während Werte >1 die Konvergenz erhöhen, und Werte <1 sie verringern. Zu geringe Werte führen allerdings dazu, dass die Konvergenz gänzlich aufgehoben wird und sogar eine Expansion entstehen kann, während zu kleine Werte zu einer frühzeitigen Konvergenz führen können. Die folgenden Variations-Experimente werden mit einem SquareEval durchgeführt mit 20 zufälligen Extrema und mit einem globalen Extremum von 100. Es wird hier der Fokus auf das Verhalten bei mehreren Extrema begrenzt, da für 1-Extremum Probleme die Spezialisierung auf Exploration generell ungeeignet ist.

In der Abbildung 6.6 ist erkennbar, dass wenige Partikel zu deutlich schlechteren Ergebnissen führen. Dies liegt daran, dass die geringe Anzahl an Partikeln wahrscheinlicher zu ungleichmäßigen Verteilungen führt und die Partikel stärker über die Evaluationen konvergieren als Varianten mit mehr Partikeln. Somit bewegt sich zwar jedes Partikel häufiger in Abhängigkeit zu den Evaluationen, allerdings nimmt mit jedem Schritt die Exploration der Partikel ab und weist exploitativeres Verhalten auf. Erhöht sich die Anzahl an Partikeln, ist die Verteilung der Partikel statistisch gesehen gleichmäßiger und die Partikel bewegen sich weniger in Abhängigkeit zu den Evaluationen. Damit sind die Partikel Bewegungen im Schnitt explorativer, da der explorative Anteil mit der Schrittmenge der einzelnen Partikel abnimmt.

Aus den Folgeuntersuchungen in Abbildung 6.7 ist zu erkennen, dass SPSO Varianten mit gleicher Partikelanzahl im Verlauf sehr ähnlich sind, wobei Varianten mit einem höheren rFactor und somit stärkerer Konvergenz generell bessere Ergebnisse liefern, bis sie ab einem spezifischen Punkt stark abfallen. Dieser Abfallpunkt erscheint früher bei höherem rFactor, was sehr gut bei 25 Partikeln in Abbildung 6.7b zu erkennen ist. Ein ähnliches Verhalten ist erkennbar bei der Partikelanzahl Variation. Wenige Partikel führen schneller zu besseren Ergebnissen, erreichen aber früher ein Plateau, ab welchem sie kaum noch bessere Ergebnisse erzielen. Beide Verhalten lassen sich damit erklären, dass ein höherer rFactor, sowie eine geringere Partikel Anzahl zu einer schnelleren Konvergenz

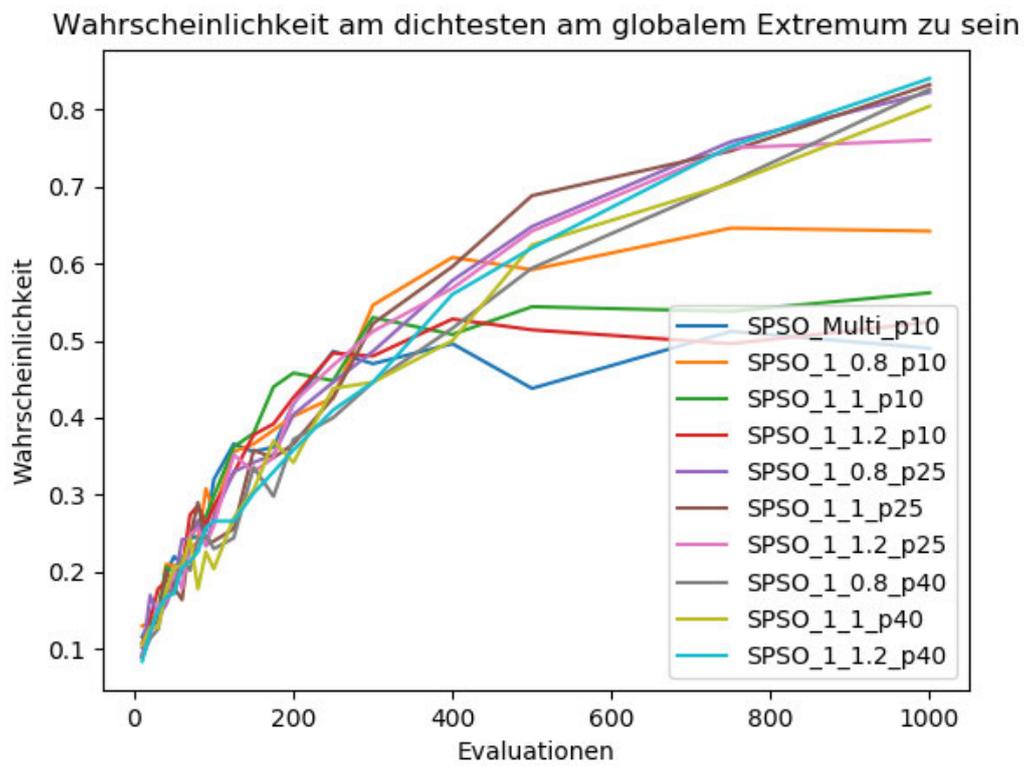
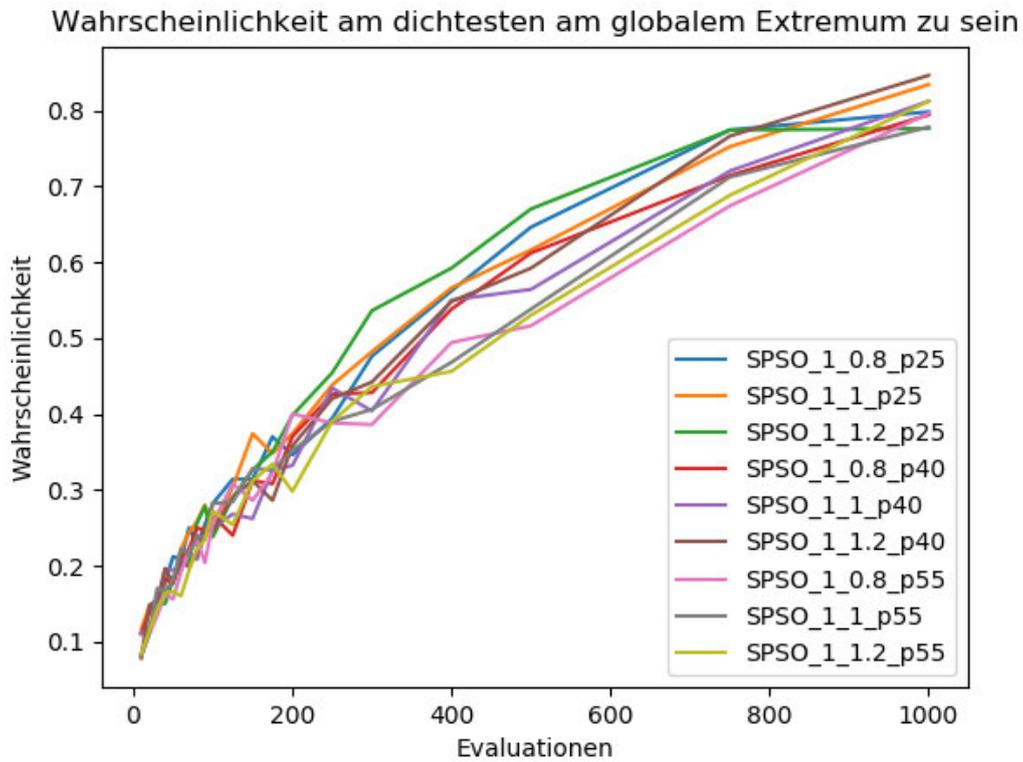
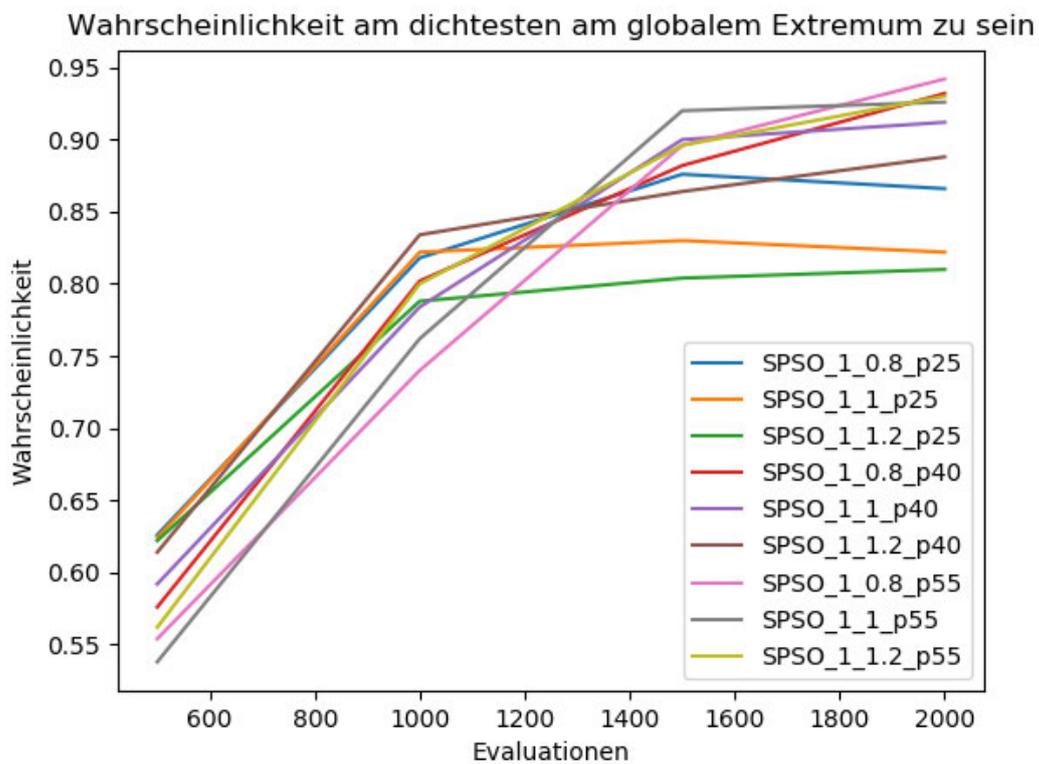


Abbildung 6.6: SPSO Varianten mit 10-40 Partikeln



(a) SPSO Varianten mit 25-55 Partikeln



(b) Erweiterung auf mehr Evaluationsen

Abbildung 6.7: Vergleich verschiedener SPSO Varianten auf ihre Wahrscheinlichkeit, am dichtesten am globalen Extremum zu sein

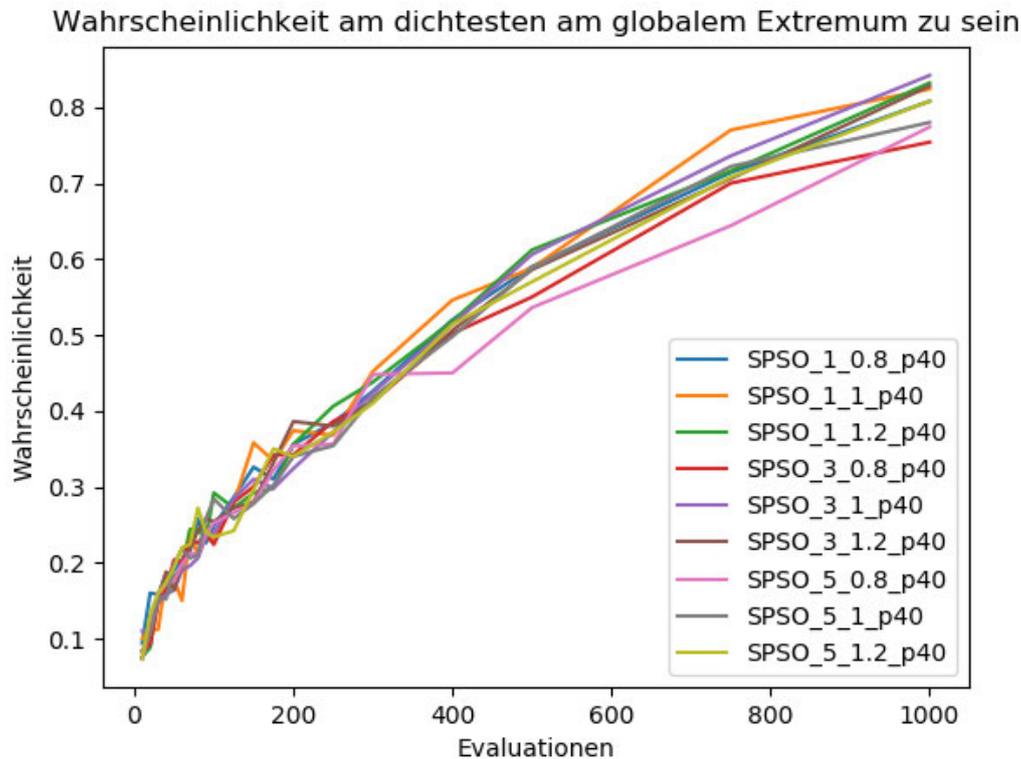


Abbildung 6.8: SPSO Varianten mit 40 Partikeln und verändertem g_{para}

über die Evaluationen führt. Die Konvergenz führt zwar anfangs zu besseren Ergebnissen, limitiert aber über längere Zeiträume hinweg das Ergebnis, da schneller von Exploration zu Exploitation gewechselt wird. Somit lässt sich für jede Anzahl an Evaluationen eine Variante finden, die jeweils das beste Ergebnis liefern. Das bedeutet für die Auswahl für den Hybriden, dass die beste Variante abhängig von der SPO Initialisierungsbedingung und der damit zusammenhängenden PSO Zurücksetzung ist. Diese Bedingungen können in statische und dynamische Bedingungen aufgeteilt werden. Bei statischen Bedingungen lässt sich von Anfang an sagen, bei wie vielen Evaluationen welche Variationen die besten sind. Bei Dynamischen ist dies schwerer vorhersagbar. Es wäre eine Anpassung des Hybriden denkbar, bei welcher dynamisch ermittelt wird, welche SPSO Variante am geeignetsten ist. Dies wird allerdings in dieser Arbeit nicht weiter verfolgt, da der Aufwand zu groß eingeschätzt wird, um dies hier in ausreichendem Umfang zu untersuchen.

In Abbildung 6.8 wird der Einfluss der Nachbarschafts Größe auf den Verlauf dargestellt. Es zeigt sich, dass Varianten mit einer kleineren Nachbarschaft zu leicht besseren Ergeb-

nissen führen. Aus diesem Grund werden für den Hybriden nur Variationen mit einem `gpara` von 1 genutzt.

Es werden für die Hybrid Variationen drei SPSO Varianten abhängig von der erwarteten Evaluationsanzahl untersucht:

- <300 Evaluationen: SPSO_1_1.2_p10
- 300-600 Evaluationen: SPSO_1_1.2_p25
- >600 Evaluationen: SPSO_1_1.2_p40

6.4 Hybrid Variation

Auf Basis der vorher ermittelten SPO und SPSO Varianten wurden mehrere Hybrid erstellt. Es wird bei den Versuchen derselbe Evaluator genutzt, welcher für die SPSO Variationen genutzt wurde.

In Abbildung 6.10 ist erkennbar, dass die meisten Varianten ein deutlich besseres Ergebnis aufweise, als die gewählte Referenz, eine Standard Implementation von SPSO, welche in der Arbeit von M.Clerc[1] angeboten wurde.

Um auf eine allgemeine Verbesserung zu prüfen, wurde der Versuch mit denselben Parametern wiederholt, allerdings diesmal mit einem `RippleEval` Evaluatoren. Das Ergebnis 6.11 zeigt, dass die Hybrid Varianten im Schnitt alle schlechter abschließen, als die SPSO Referenz. Dies deutet darauf hin, dass die Verbesserungen stark zugeschnitten sind auf eine spezielle Art von Evaluierungsfunktion und für jede eine eigene Auswahl an SPO und PSO Varianten am geeignetsten sind. Es ist hier nicht ausgeschlossen, dass es eine Konfiguration gibt, die für alle Evaluierungsfunktionen die besten Ergebnisse liefert, dies wird allerdings nicht erwartet, da es bei Metaheuristiken im Allgemeinen keine generell beste Variante gibt, sondern verschiedene Metaheuristiken für spezifische Evaluierungsfunktions Arten besser geeignet sind.

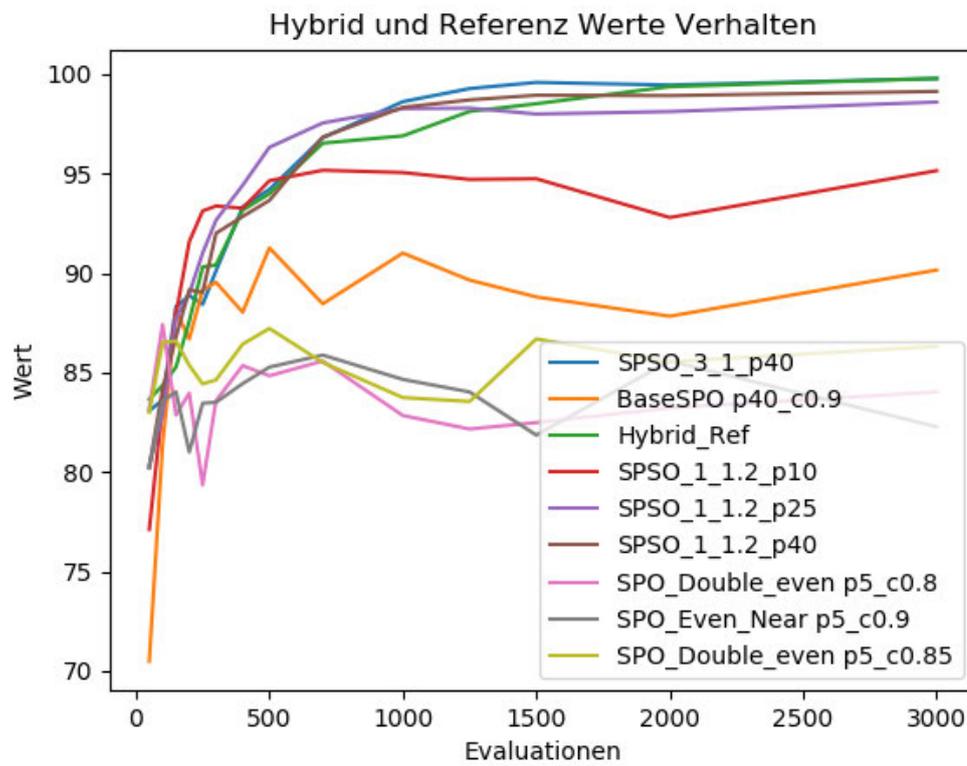


Abbildung 6.9: Werte Verhalten von einfachen Hybriden und Referenzen

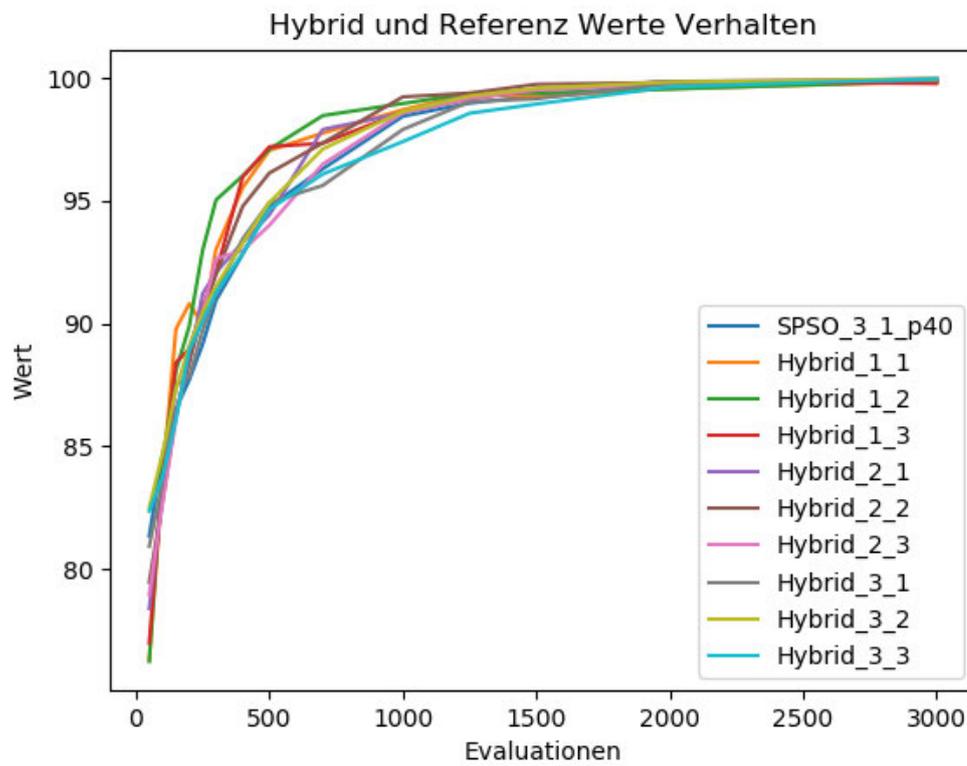


Abbildung 6.10: Hybrid Werte Verhalten bei SPO Initialisierung nach fester Zyklen Anzahl

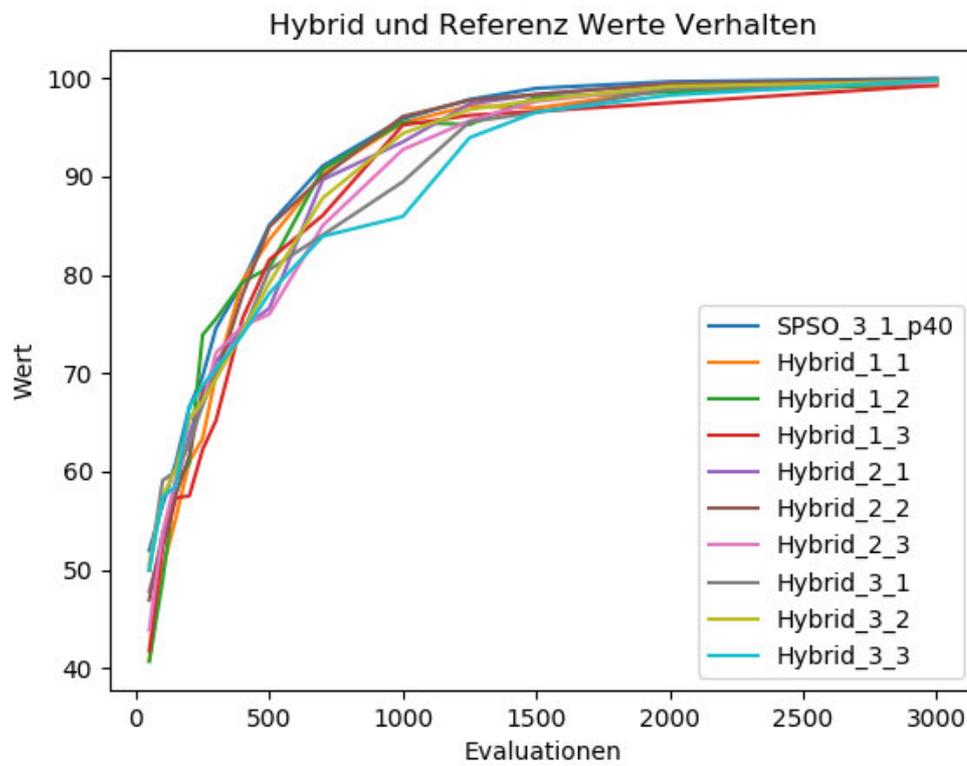


Abbildung 6.11: Hybrid Werte Verhalten bei SPO Initialisierung nach fester Zyklen Anzahl

7 Auswertung und Verbesserungen

Anhand der Experimente ist zu sehen, dass die Hybridisierung in der Lage ist, bessere Ergebnisse zu liefern als die einzelnen PSO und SPO Varianten, ohne dabei zu große Änderungen vornehmen zu müssen. Im Allgemeinen lässt sich beobachten, dass die Hybridisierung sehr davon profitiert, geringere Partikel Mengen zu nutzen. Durch das Einschränken auf ein mögliches Extremum Gebiet durch PSO ergeben sich simplere Gebiete auf die SPO angepasst werden kann. Im Ideal Fall wird innerhalb kürzester Zeit auf ein Gebiet begrenzt, welches nur ein Extremum enthält, wodurch eine sehr starke Fokussierung auf Exploitation möglich ist. Diese kann zwar extrem schnell präzise Ergebnisse liefern, führt allerdings bei mehreren Extrema auf Grund mangelnder Exploration dazu, dass mit höherer Wahrscheinlichkeit zu lokale Extrema, anstelle von globalen konvergiert wird. Während die Fokussierung von SPO erfolgreich gelungen ist, weisen die Ergebnisse der Anpassung von SPSO nur relativ geringere Verbesserungen auf. Es war zwar generell möglich für unterschiedliche Mengen an Evaluationen bessere Ergebnisse zu erzielen, diese weisen allerdings meist nur geringe Verbesserungen auf. Trotzdem hat es sich gezeigt, dass die Hybridisierung in der Lage ist bessere Ergebnisse zu liefern. Eine Auffälligkeit hierbei ist, dass bei geringen Partikel Mengen von SPO und PSO die einfache Hybridisierung beider Verfahren eher Vorteile aufweist als bei größeren Mengen. Dies ist in Abbildungen 6.3 und 6.9 zu beobachten und lässt sich damit erklären, dass abhängig von der Evaluationsfunktion Verfahren geringerer Partikelanzahl mit einer gewissen Wahrscheinlichkeit nicht in der Lage sind das globale Maximum zu finden. In den Abbildungen ist erkennbar, dass die Verläufe nicht zum Extremumwert 100 konvergieren, sondern bereits vorher aufhören zu steigen. Die Hybridisierung ermöglicht es, diese Hürde mit Hilfe der Ausschlusszonen und dem Zurücksetzen von PSO zu überwinden. Wird nun allerdings die Partikelanzahl erhöht, wird diese Hürde immer kleiner, da die Wahrscheinlichkeit, das globale Extremum zu finden, immer größer wird. Die Vorteile des Hybriden werden zudem dadurch vermindert, dass die Hybrid Implementierung abhängig von den Zyklen ist. Dies bedeutet, dass bei mehr Partikeln die Zyklen länger werden und somit über die Evaluationen betrachtet der Hybrid immer weniger zum Tragen kommt, was durch

die allgemein wenigen benötigten Evaluierungen dieser Algorithmen noch verstärkt wird. Dieses Zyklen basierte Verhalten erschwert es ebenfalls, SPO und PSO Varianten mit stark unterschiedlichen Partikel Mengen zu nutzen, da in jedem Hybrid Zyklus PSO und SPO je einen Zyklus durchlaufen, was durchaus problematisch ist, da die Experimente gezeigt haben, dass Exploitation geringe Partikel Zahlen bevorzugt, während Exploration mehr Partikel für gute Ergebnisse braucht. Es entsteht somit ein festes Verhältnis zwischen Exploration und Exploitation, tendenziell zu Gunsten der Exploration. Zwei mögliche Anpassungen wären entweder PSO und SPO pro Hybrid Zyklus mit mehreren und teils nicht vollen Zyklen aufzurufen, sodass ein gewünschtes Verhältnis an Evaluationen pro Zyklus entsteht, oder für beide Verfahren die Zyklen Anzahl pro Hybrid Zyklus anzupassen, sodass sie ungefähr das gewünschte Verhältnis haben. Ersteres bedeutet, dass SPO und PSO dies unterstützen müssen. Dies würde den Hybriden in seiner Flexibilität einschränken, da es umständlicher wird, die genutzten Algorithmen für die Integration anzupassen. Zweiteres erfordert keine zusätzliche Anpassung der genutzten Algorithmen, allerdings müssen entweder bei der Verhältniswahl Kompromisse eingegangen werden, oder so viele zusätzliche Zyklen während eines Hybrid Zyklus durchgeführt werden, dass die Hybridisierung an Effektivität abnimmt. Ein weiteres Problem kann auftreten durch zu hohe Abdeckung oder Größe von Ausschlusszonen. Wenn einzelne Zonen zu groß werden, wird es schwerer, effektive exploitative Algorithmen zu finden, die mit der dadurch erhöhten Wahrscheinlichkeit auf mehrere Extrema pro Zone kompatibel sind. Zusätzlich kann es dadurch schneller zu einer totalen Abdeckung kommen, bei der Exploration nicht mehr möglich ist. Dies ist hauptsächlich relevant, wenn sehr häufig neue SPO Cluster initialisiert werden oder die Laufzeit sehr lang ist. Wie bereits vorher erwähnt, wäre eine hierfür geeignete Lösung eine kontinuierlich schrumpfende Ausschlusszonen Größe. Dies erlaubt es unbegrenzt lange den Algorithmus laufen zu lassen und immer wieder neue Gebiete zum Erkunden zu haben, auch wenn diese vorher ausgeschlossen worden sind. Durch das schrumpfen der Ausschlusszonen würde ebenfalls das Problem mit der Wahl der Ausschlusszonen Größe verringert werden, da Extrema, die zu dicht beieinander liegen und so in die selbe Ausschlusszone gefallen sind, wieder neu entdeckt werden können.

8 Abschluss

8.1 Zusammenfassung

In dieser Arbeit wurde eine Art der Hybridisierung von zwei Metaheuristiken am Beispiel von SPO und SPSO vorgestellt und bewertet, ob es sich lohnt diese weiter zu untersuchen. Es wurde SPSO so konfiguriert, dass es einen starken explorativen Charakter hat, sodass potenzielle globale Extrema möglichst schnell gefunden werden, ohne diese aber genau zu bestimmen. Für die Exploitation wurden SPO Varianten bestimmt, welche in der Lage sind, in Bereichen mit nur einem Extremum in kürzester Zeit beste Ergebnisse zu liefern. Die Hybride aus diesen Varianten zeigen, dass die Hybridisierung in der Lage ist, für bessere Ergebnisse zu sorgen. Dies ist besonders dann der Fall, wenn das Explorative Verfahren dazu neigt, zu lokalen Extrema zu konvergieren, da durch die Hybridisierung kontinuierlich neue Extrema gefunden werden. Laufzeit Untersuchungen haben ebenfalls gezeigt, dass die Hybridisierung die Laufzeit des explorativen Verfahrens multiplikativ mit der Anzahl der Ausschlusszonen beeinflusst, allerdings ist dieser Effekt relativ klein und hat erst bei längeren Laufzeiten einen signifikanten Einfluss. Des Weiteren ist es möglich, aus dem Zustand des Hybriden nicht nur ein bestes Extremum zu bestimmen, sondern es ist möglich eine Vielzahl an Extrema zu erhalten.

8.2 Fazit

Es wurde erfolgreich gezeigt, dass die Hybridisierung von PSO und SPO in der Lage ist, bessere oder gleichwertige Ergebnisse zu liefern als die einzelnen Verfahren. Es hat sich sogar gezeigt, dass es möglich ist, bessere Ergebnisse zu erzielen als Varianten, die an sich für diese Probleme geeignet wären. Diese Ergebnisse deuten darauf hin, dass diese Art der Hybridisierung in der Lage ist, mehrere metaheuristische Verfahren mit einander zu verknüpfen und dabei positive Einflüsse erzeugt.

8.3 Ausblick

In dieser Arbeit wurde nur mit je einem Verfahren für Exploration und Exploitation, sowie nur mit einer begrenzten Anzahl an Evaluierungsfunktionen gearbeitet. Aus diesem Grund sind die Ergebnisse dieser Arbeit nur ein kleiner Einblick in das gesamte Verhalten der Hybridisierung. Es sind weitere und größere Experiment Reihen von Nöten, um einen generell positiven Effekt bestimmen zu können, und zu untersuchen, welche Arten von Metaheuristiken für die Hybridisierung am geeignetsten sind. Eine Untersuchung für welche Arten von Evaluierungsfunktionen welche Hybridisierungen geeignet sind, wäre zwar interessant, allerdings wäre die Untersuchung sehr Umfangreich, da einzelne Metaheuristiken besser für unterschiedliche Evaluierungsfunktionen sind, und Untersuchungen zur Eignung einer Hybridisierung von mehreren unterschiedlichen Metaheuristiken dem Autor nicht bekannt sind. Eine weitere Untersuchung, die sich lohnen könnte, wäre die Untersuchung einer Anpassung des Hybriden mit zwei stark Exploitativen Verfahren. Es ist zu erwarten, dass diese Variante zwar langsamer ist, verlässlich ein globales Extremum zu finden, jedoch wäre es damit möglich eine Vielzahl an Extrema in kurzer Zeit zu bestimmen. Diese Ergebnisse könnten helfen, die Extrema Verteilung einer Evaluierungsfunktion zu bestimmen und zu untersuchen. Es zeigt sich, dass noch einiges an Bedarf an weiteren Untersuchungen zu diese Hybridisierung besteht und diese auf Grund der Ergebnisse dieser Arbeit durchaus lohnenswert sein könnten.

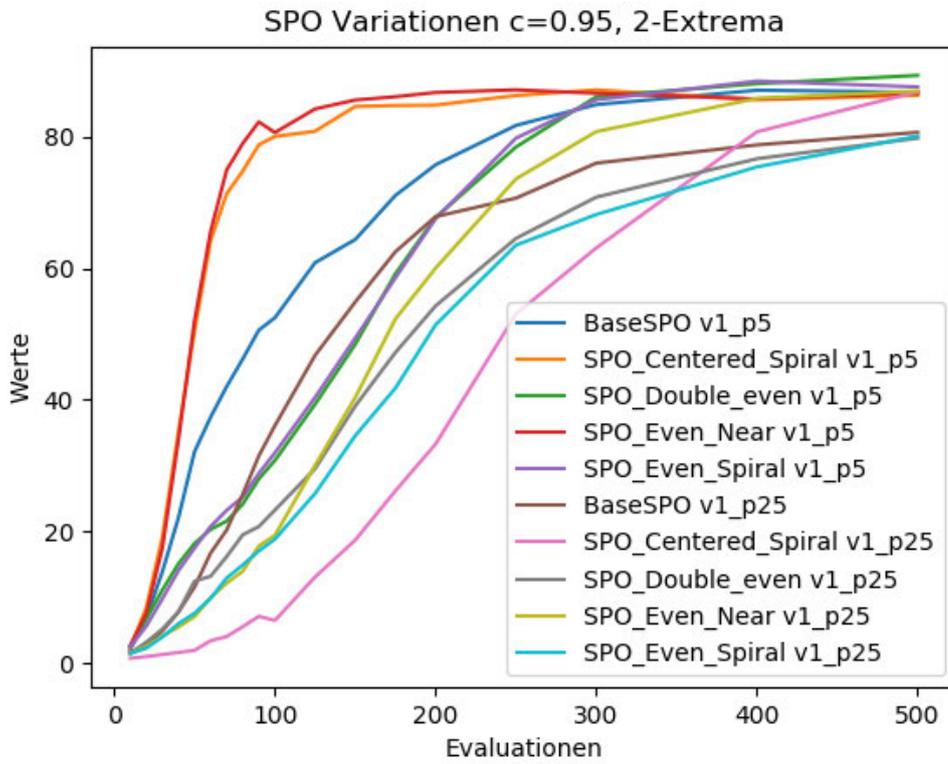
Literaturverzeichnis

- [1] CLERC, Maurice: *Standard Particle Swarm Optimisation*. 2012. – URL <https://hal.archives-ouvertes.fr/hal-00764996>. – Zugriffsdatum: 2020-08-17
- [2] KENNEDY, J. ; EBERHART, R.: Particle swarm optimization. In: *Proceedings of ICNN'95 - International Conference on Neural Networks* Bd. 4, 1995, S. 1942–1948 vol.4
- [3] KENNEDY, J. ; MENDES, R.: Population structure and particle swarm performance. In: *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No.02TH8600)* Bd. 2, 2002, S. 1671–1676 vol.2
- [4] MONTES DE OCA, M. A. ; PENA, J. ; STUTZLE, T. ; PINCIROLI, C. ; DORIGO, M.: Heterogeneous particle swarm optimizers. In: *2009 IEEE Congress on Evolutionary Computation*, 2009, S. 698–705
- [5] POLI, Riccardo: Analysis of the Publications on the Applications of Particle Swarm Optimisation. In: *Journal of Artificial Evolution and Applications* 2008 (2008), Feb, S. 685175. – URL <https://doi.org/10.1155/2008/685175>. – ISSN 1687-6229
- [6] SHI, Y. ; EBERHART, R.: A modified particle swarm optimizer. In: *1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98TH8360)*, 1998, S. 69–73
- [7] TALBI, E.-G.: A Taxonomy of Hybrid Metaheuristics. In: *Journal of Heuristics* 8 (2002), Sep, Nr. 5, S. 541–564. – URL <https://doi.org/10.1023/A:1016540724870>. – ISSN 1572-9397
- [8] TAMURA, Kenichi ; YASUDA, Keiichiro: Spiral Dynamics Inspired Optimization. In: *Journal of Advanced Computational Intelligence and Intelligent Informatics* 15 (2011), Nr. 8, S. 1116–1122

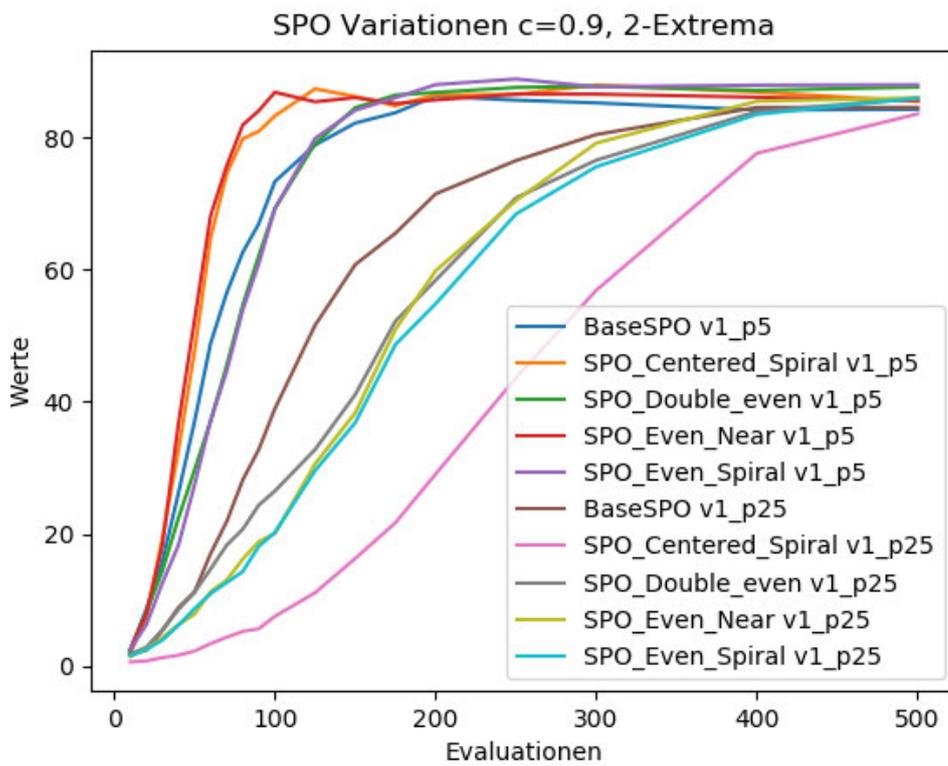
A Anhang

A.1 Versuche zu den SPO Varianten

A.1.1 SPO Variation mit 2-Extrema

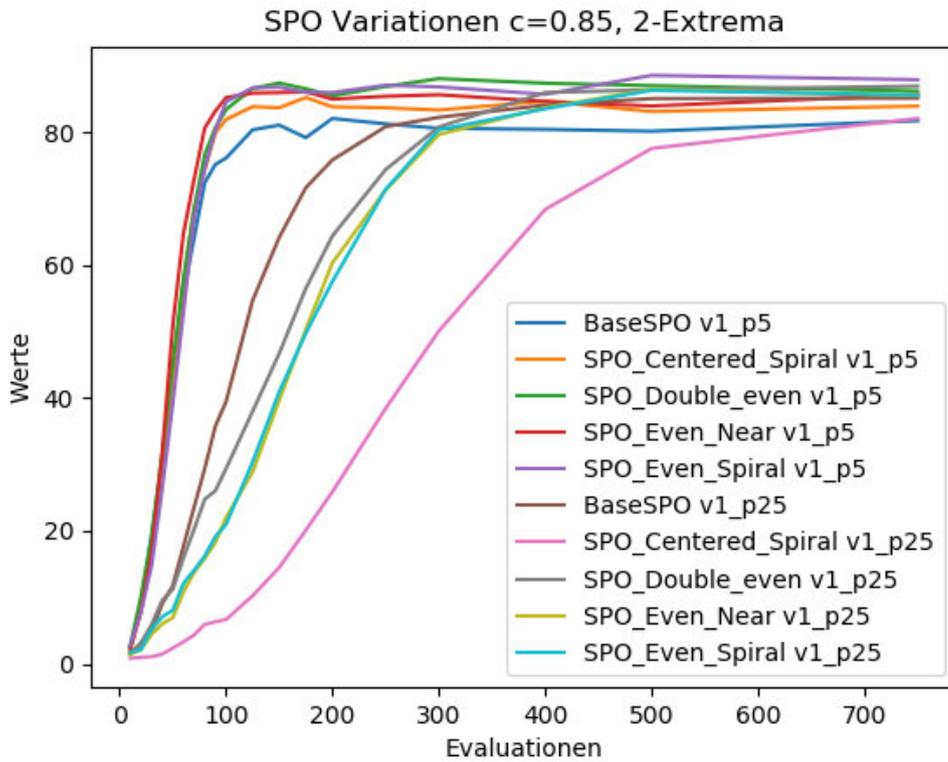


(a) SPO Variationen $c=0.95$, 2 Extrema

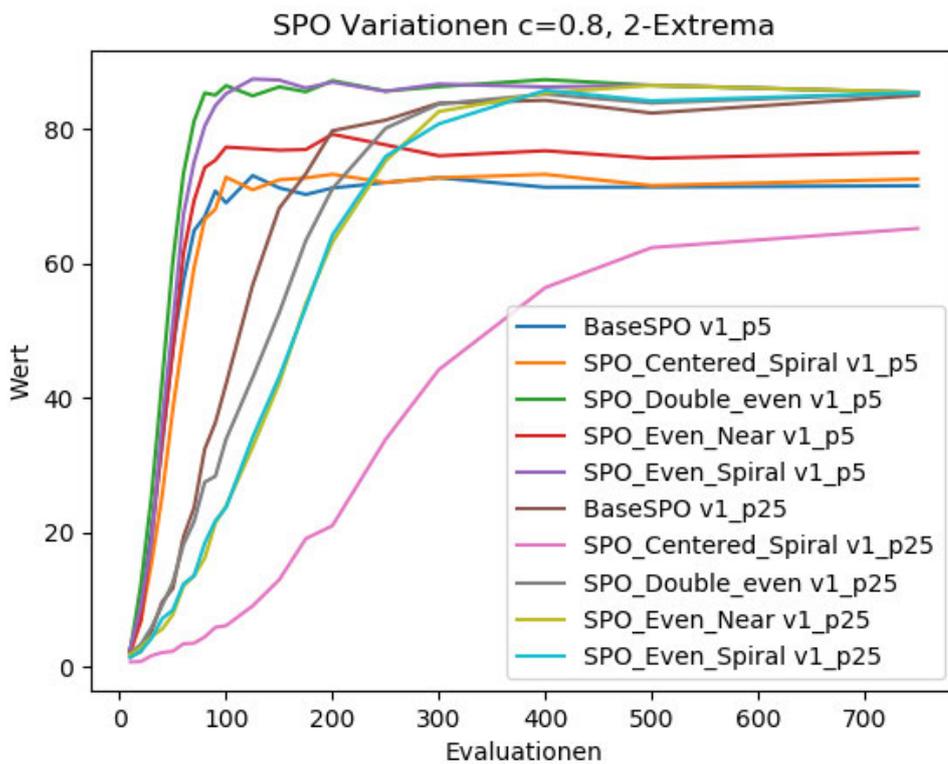


(b) SPO Variationen $c=0.9$, 2 Extrema

Abbildung A.1: SPO Variationen mit Konvergenzfaktor 0.95 und 0.9 mit SquareEval Evaluator und 2 Maxima



(a) SPO Variationen $c=0.85$, 2 Extrema



(b) SPO Variationen $c=0.8$, 2 Extrema

Abbildung A.2: SPO Variationen mit Konvergenzfaktor 0.85 und 0.8 mit SquareEval Evaluator und 2 Maxima

Erklärung zur selbstständigen Bearbeitung einer Abschlussarbeit

Gemäß der Allgemeinen Prüfungs- und Studienordnung ist zusammen mit der Abschlussarbeit eine schriftliche Erklärung abzugeben, in der der Studierende bestätigt, dass die Abschlussarbeit „— bei einer Gruppenarbeit die entsprechend gekennzeichneten Teile der Arbeit [(§ 18 Abs. 1 APSO-TI-BM bzw. § 21 Abs. 1 APSO-INGI)] — ohne fremde Hilfe selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel benutzt wurden. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich zu machen.“

Quelle: § 16 Abs. 5 APSO-TI-BM bzw. § 15 Abs. 6 APSO-INGI

Erklärung zur selbstständigen Bearbeitung der Arbeit

Hiermit versichere ich,

Name: _____

Vorname: _____

dass ich die vorliegende Bachelorarbeit – bzw. bei einer Gruppenarbeit die entsprechend gekennzeichneten Teile der Arbeit – mit dem Thema:

Erarbeitung und Untersuchung eines Hybriden aus PSO und SPO.

ohne fremde Hilfe selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich gemacht.

_____  _____
Ort Datum Unterschrift im Original