# Anpassungen von Suchheuristiken für ereignisdiskrete Modelle

**Dissertation**

zur Erlangung des Grades eines

**D o k t o r s  d e r  N a t u r w i s s e n s c h a f t e n**

der Technischen Universität Dortmund
an der Fakultät für Informatik
von

**Dennis Müller**

Dortmund
2010

Dennis Müller
Lehrstuhl IV - Modellierung und Simulation
Fakultät für Informatik
Technische Universität Dortmund
August-Schmidt-Str. 12
44227 Dortmund

| | |
|---|---|
| Tag der mündlichen Prüfung | 07.07.2010 |
| Dekan | Prof. Dr. Peter Buchholz |
| Gutachter | Prof. Dr. Peter Buchholz,<br>Prof. Dr. Günter Rudolph |
| Vorab veröffentlichter Eigenanteil | ca. 30% |

# Danksagung

An dieser Stelle bedanke ich mich bei Herrn Prof. Dr. Peter Buchholz für die Betreuung dieser Dissertation, die Bereitstellung hervorragender Arbeitsbedingungen sowie die vielen gewährten Freiräume. Herrn Prof. Dr. Günter Rudolf danke ich für die Übernahme des Zweitgutachtens. Zusätzlich möchte ich mich bei allen Mitarbeitern des Lehrstuhls IV für die freundliche Arbeitsatmosphäre bedanken.

Die vorliegende Arbeit entstand am Lehrstuhl IV der Informatik der Technischen Universität Dortmund. Teile dieser Arbeit sind im Rahmen des Sonderforschungsbereichs 559 „Modellierung großer Netze in der Logistik“ der Deutsche Forschungsgemeinschaft sowie der Nachwuchsforschergruppe „Ganzheitliche Modellierung ereignisorientierter Systeme“ des Dortmunder Forschungsbands für Modellbildung und Simulation entwickelt worden. Mein Dank gilt damit auch der aus den Forschungsgruppen entstandenen Unterstützung.

# Inhaltsverzeichnis

# Einleitung

Aufgrund der in den letzten Jahrzehnten stark gestiegenen Rechenleistungen aktueller Computersysteme und der Erschließung weiterer Einsatzgebiete durch computergestützte Systeme können heute deutlich komplexere Aufgaben automatisiert durchgeführt werden. Eine dieser Aufgaben ist die Planung oder Modifizierung komplexer, technischer Systeme, zu denen z.B. Logistiknetze gehören, wie sie im Sonderforschungsbereich 559 [Kuh98] der Deutschen Forschungsgemeinschaft untersucht wurden. In Zeiten der Globalisierung können wirtschaftlich arbeitende Logistiknetze nicht mehr auf kleine Bereiche beschränkt werden, sondern müssen landes-, kontinenten- oder weltweite Interaktionen berücksichtigen. Dabei müssen zu jedem Zeitpunkt Anforderungen wie Verlässlichkeit, Kostengünstigkeit und Leistungsfähigkeit berücksichtigt werden. Aufgrund der Vielzahl der Variationsmöglichkeiten bei der Konfiguration ist es in der Regel nicht mehr möglich, eine manuelle Auswahl durchzuführen, so dass sowohl die Analyse als auch die Auswahl modellbasiert und computergestützt durchgeführt wird.

Zur Modellierung solcher Systeme werden in vielen Fällen ereignisdiskrete Modelle eingesetzt. Solche Modelle zeichnen sich dadurch aus, dass sich der Zustand eines Systems nur zu bestimmten Zeitpunkten ändert, wobei der Zustandsraum in einigen Fällen endlich aber sehr groß ist, in anderen Fällen aber unbeschränkt oder sogar überabzählbar sein kann. Logistiknetze werden oft mit Hilfe von Petri- oder Warteschlangenetzen modelliert, da diese einen hohen Detaillierungsgrad ermöglichen und effiziente Analyseverfahren existieren. Für eingeschränkte Modellklassen sind darüber hinaus weitere Analyseverfahren entwickelt worden, die jedoch in ihrer Einsetzbarkeit stark beschränkt sind [Arn06].

Mit diesen Analysemethoden können verschiedene Modelle bezüglich bestimmter Kriterien bewertet werden. Häufig besteht das Ziel darin, eine optimale Konfiguration aus einer Menge möglicher Konfigurationen auszuwählen und anschließend umzusetzen. Da die Menge der möglichen Konfigurationen in der Regel sehr groß und die Unterschiede in den Konfigurationen in vielen Fällen sehr gering sind, ist es nicht sinnvoll die beste Konfiguration von Hand zu ermitteln. Daher sollen computergestützte Verfahren entwickelt oder bestehende angepasst werden, um sie in den beschriebenen Situationen nutzen zu können. Im Rahmen des SFB 559 wurden verschiedene Verfahren entwickelt, die sich auf spezielle Aufgabenstellungen beim Entwurf von Logistiknetzen konzentrieren. Dabei werden z.B. Fragestellungen wie die der optimalen Anzahl von Verteilzentren und deren Position oder nach einer optimalen Struktur beantwortet. Bei vielen anderen Fragestellungen ist die Struktur des

Modells bereits festgelegt und es stehen „nur“ noch mehrere Varianten eines Modells zur Auswahl. Eine typische Situation, in der eine solche Fragestellung auftritt, ist die Frage nach der optimalen Konfiguration eines Rechners für eine bestimmte Aufgabe. Der grundsätzliche Aufbau des Rechners hat sich über Jahre bewährt, so dass ausschließlich zu entscheiden ist, welche Leistungsdaten die benötigten Komponenten erfüllen müssen. Zu diesen gehören z.B. durchschnittliche Durchsatzraten von Festplatten. Da hier ausschließlich Parameter variiert werden, können diese Fragestellungen als Optimierung einer Funktion betrachtet werden. Automatisierte Verfahren zur Optimierung haben eine lange Geschichte in der Informatik und dementsprechend breit gefächert sind deren Einsatzgebiete.

Die Untersuchung und Anpassung von Optimierungsverfahren für ereignisdiskrete Modelle ist dabei ebenfalls kein neues Forschungsgebiet. Aufgrund des hohen Rechenbedarfs solcher Verfahren war und ist die Menge der einsetzbaren Verfahren jedoch beschränkt - es müssen Methoden entwickelt werden, die mit den aktuell vorhandenen Ressourcen gute Ergebnisse produzieren. Darüber hinaus sind viele Verfahren wie z.B. RSM [MM02] in ihrer allgemeinen Beschreibung zwar für ereignisdiskrete Modelle geeignet, jedoch nicht vollständig automatisiert, so dass für den Anwender eine genaue Kenntnis der eingesetzten Methoden notwendig ist. Ziel dieser Arbeit ist es daher bereits bekannte Verfahren so anzupassen, dass sie vollständig automatisiert ablaufen, vorhandene Ressourcen sinnvoll nutzen und die Eigenschaften ereignisdiskreter Modelle bzw. deren Analysemethoden berücksichtigen. Die Ergebnisse dieser Arbeit fließen dabei in die Software OPEDo [BMS05] (siehe Anhang B) sowie in den SFB 559 [ABM09] ein.

Die bisher beschriebenen Fragestellungen und Probleme sollen in dieser Arbeit genauer untersucht werden. In den Kapiteln 1 und 2 werden die für die weitere Arbeit benötigten Grundlagen bezüglich ereignisdiskreter Modelle, stochastischer Prozesse und deren Analysemethoden erklärt. Kapitel 3 befasst sich mit der Betrachtung von veränderbaren Modellen als Funktionen und führt für die weiteren Untersuchungen einige Benchmarkmodelle und -funktionen ein. In Kapitel 4 werden Experimentdesigns als Untersuchungsmethode für Funktionen betrachtet, mit deren Hilfe die in Kapitel 5 vorgestellten Verfahren Approximationen der zu Grunde liegenden Funktionen ermitteln. In Kapitel 6 wird der Begriff der Optimierung im Sinne dieser Arbeit genauer eingegrenzt und einige bekannte Suchheuristiken vorgestellt. Kapitel 7 führt die Optimierung mit Kriging Metamodellen ein, die zentraler Bestandteil dieser Arbeit ist. In Kapitel 8 werden hierarchische Kriging Metamodelle vorgestellt, mit deren Hilfe je nach Konfiguration der Rechenaufwand gesenkt oder die Güte der Approximation erhöht werden kann. In Kapitel 9 wird ein hybrides Verfahren vorgestellt, dass die Vorteile von globalen und lokalen Suchheuristiken vereinen soll. Die Einsetzbarkeit von Kriging Metamodellen in der Optimierung von Simulationsmodellen wird abschließend in Kapitel 10 betrachtet.

# 1. Grundlagen

Problemstellungen aus dem Bereich der Optimierung treten in vielen unterschiedlichen Kontexten auf. Die Optimierung von Systemen ist daher in vielen Bereichen wie zum Beispiel der Informatik, der Logistik oder auch der Kommunikationstechnik eine häufige Aufgabe. Die Entwicklung neuer Verfahren wird auf wissenschaftlicher Seite vor allem in den Bereichen Mathematik und Informatik sowie mit oft stärker wirtschaftlichem Fokus im Operations Research angesiedelt. Anwendung finden die entwickelten Verfahren hingegen in vielen weiteren wissenschaftlichen Themengebieten und haben auch für viele Unternehmen eine große Bedeutung.

Die häufigsten Fragestellungen aus den Bereichen der Informatik, der Logistik sowie der Kommunikationstechnik beziehen sich darauf, wie man ein bestimmtes Ziel mit einem optimalen Kosten-Nutzen-Verhältnis mit einem System erreichen kann. Sowohl für den Nutzen als auch für die Kosten existieren dabei in der Regel Bewertungsfunktionen, die miteinander verknüpft sein können. In vielen Fällen ist darüber hinaus die grundlegende Struktur des Systems bereits vorgegeben. Beispielsweise wird bei der Planung eines neuen Webservers der allgemeine Aufbau von Servern nicht in Frage gestellt. Daher reduzieren sich die Freiheitsgrade darauf zu entscheiden, welche Ressourcen, Strategien, etc. eingesetzt werden.

Die meisten aktuellen oder aktuell geplanten Systeme zeichnen sich durch eine hohe Komplexität aus, wodurch eine rein auf Erfahrungswerten basierte Planung zu aufwändig wird oder bereits nicht mehr möglich ist. Darüber hinaus führt die rasante technische Entwicklung dazu, dass zu vielen Systemen noch keine ausreichenden Erfahrungswerte existieren. Daraus folgt, dass die computergestützte Analyse solcher Systeme ein sinnvoller Weg für die Bearbeitung solcher Fragestellungen ist. Da jedoch nicht beliebige Systeme computergestützt analysiert werden können, müssen diese im Allgemeinen zuerst in ein adäquates Modell überführt werden.

## 1.1. Modellarten

Für unterschiedliche Analyseziele und Systeme wurde eine Vielzahl von Modellarten entwickelt. Viele dieser Modellarten sind für die bisher beschriebenen Fragestellungen jedoch nicht oder nicht sinnvoll einsetzbar. Die folgende Einführung umfasst daher nur die für die restliche Arbeit wichtigsten Modellarten wie z.B. Warteschlangennetze.

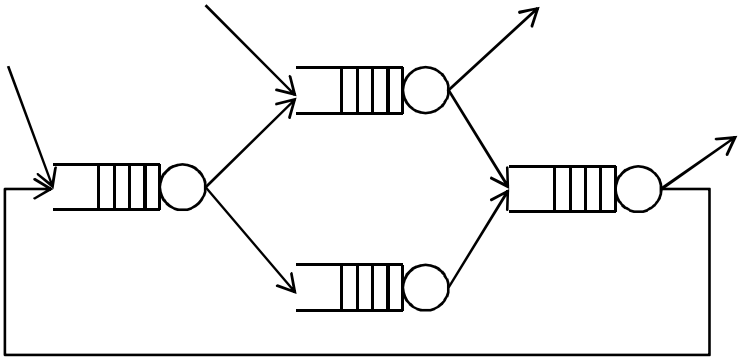


Abbildung 1.1.: Einfaches offenes Warteschlangenetz mit vier Stationen, zwei Eingängen und zwei Ausgängen

### 1.1.1. Warteschlangennetze

Warteschlangennetze [LZGS84] sind ein relativ einfacher, aber trotzdem mächtiger Formalismus und stehen im Mittelpunkt der Dissertation von Arns [Arn06], an der sich diese Beschreibung orientiert. Ein Warteschlangennetz (siehe Abbildung 1.1) besteht aus Kunden, $l$ Pfaden und $m$ Warteschlangen, die auch Bedieneinrichtungen oder Stationen genannt werden. Jede Station $w_i$ mit $i = 1, \ldots, m$ besteht aus einem Warteraum für die Kunden mit endlicher oder unendlicher Kapazität und $k_i$ Bedienern, die mit einer Bedienrate $s_i$ und einer Bedienstrategie $b_i$ die einzelnen Bedienwünsche der Kunden abarbeiten. Die Bedienstrategie legt dabei fest, in welcher Reihenfolge die Kunden aus dem Warteraum entnommen werden. Die Stationen sind über Pfade miteinander verbunden, über die die Kunden nach ihrer Abarbeitung zeitlos von einer Station zur nächsten gelangen. Die Auswahl der Pfade wird über sogenannte Routing-Wahrscheinlichkeiten festgelegt. Die Bedienwünsche der Kunden sind durch kontinuierliche Zufallsvariablen beschrieben, durch die die zeitliche Inanspruchnahme der Bediener festgelegt wird. Die zugrunde liegende Verteilung dieser Zufallsvariablen ist von der Maschine und dem Typ des Kunden abhängig.

Im Allgemeinen wird zwischen offenen und geschlossenen Warteschlangennetzen unterschieden. Bei offenen Warteschlangennetzen treffen neue Kunden aus der Umgebung an bestimmten Stationen ein, wobei die Eintrittszeitpunkte durch kontinuierliche Zufallsvariablen festgelegt sind. Darüber hinaus können Kunden das System an bestimmten Stationen wieder verlassen. Bei geschlossenen Warteschlangennetzen können Kunden das System weder betreten noch verlassen, so dass die Anzahl der Kunden im Netz konstant ist.

Warteschlangennetze und deren Erweiterungen sind die Grundlage für die Modellarten vieler Simulationswerkzeuge. Eine eingeschränkte Unterklasse kann auch auf

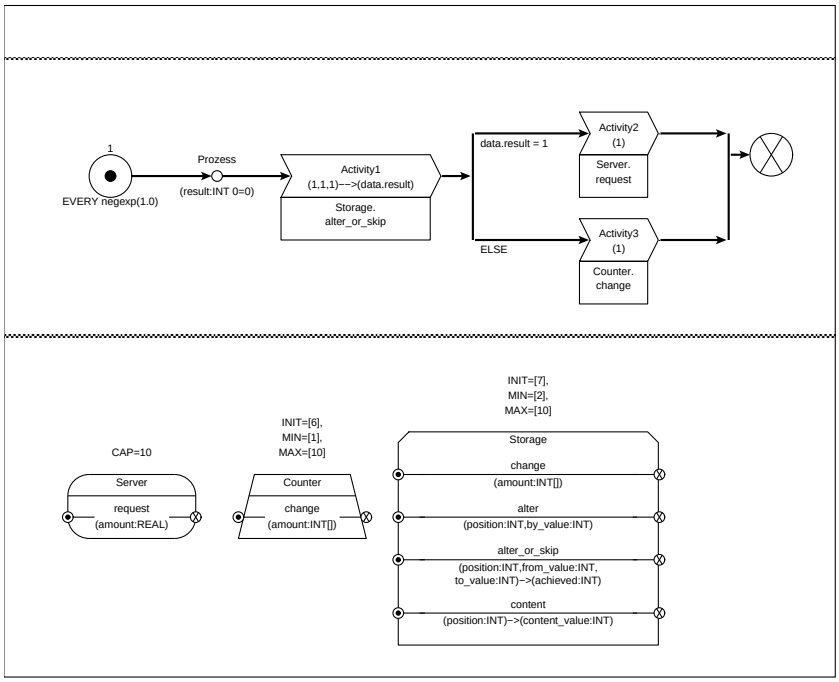



Abbildung 1.2.: Einfaches ProC/B-Modell: Die Ablaufbeschreibung befindet sich im oberen Teil und führt von einer Quelle (links) über mehrere Aktivitäten und eine Verzweigung zu einer Senke (rechts). Die verwendeten Komponenten mit den jeweils zur Verfügung gestellten Funktionen sind im unteren Bereich dargestellt.

Markov-Prozesse abgebildet und damit numerisch oder analytisch analysiert werden. Bei der Optimierung von Warteschlangenmodellen werden vor allem die Anzahl Bediener, die Bedienrate oder die Verzweigungswahrscheinlichkeiten so modifiziert, dass bestimmte Leistungsgrößen wie z.B. Durchsätze, Auslastungen oder Verweilzeiten maximiert bzw. minimiert werden.

### 1.1.2. Prozessketten

Prozessketten stellen einen in der Logistik weit verbreiteten Modellierungsformalismus dar, der in [Kuh95] formalisiert wurde. Diese Formalisierung und deren Semantik sind als ProC/B-Paradigma [BBS03] im ProC/B-Toolset [BBF$^+$02] umgesetzt. Die folgende Beschreibung orientiert sich an [BBK09]. Zum besseren Verständnis der folgenden Beschreibung ist ein einfaches ProC/B-Modell in Abbildung 1.2 aufgeführt.

Eine Prozesskette beschreibt den allgemeinen Ablauf einer Klasse von Prozessen, indem sie die einzelnen Aktivitäten und deren Verknüpfung beschreibt. Sie beginnt in einer Quelle, an der die Prozesse der beschriebenen Klasse generiert werden. Die Quelle enthält Informationen, zu welchen Zeitpunkten wie viele Prozesse erstellt werden sollen, wobei die Zeitpunkte in Form von Zwischenankunftszeiten durch eine kontinuierliche Zufallsvariable dargestellt werden. Das Ende einer Prozesskette bildet eine Senke. Die Aktivitäten werden durch Prozesskettenelemente (in Abbildung 1.2 **Activity1** bis **Activity3**) dargestellt. Prozesskettenelemente können einen oder mehrere direkte Nachfolger haben, so dass parallele oder alternative Aktivitäten modelliert werden können. In Abbildung 1.2 ist z.B. nach **Activity1** eine alternative Verzweigung modelliert. Prozesskettenelemente können wahlweise eine reine Verzögerung enthalten, die auch durch eine Zufallsvariable festgelegt werden kann, oder eine Ressource nutzen. Welche Ressource wie genutzt wird, wird im unteren Teil des Prozesskettenelements dargestellt.

Die nutzbaren Ressourcen werden unterhalb des zeitlichen Ablaufs eines Prozesses modelliert. Zu diesen gehören vor allem die Elemente **Server** und **Counter**. Ein **Server** kann von einer Aktivität angefordert werden und wird damit für eine bestimmte Zeit von dieser Aktivität belegt. Sollte der **Server** bereits belegt sein, warten die Aktivitäten in einer Warteschlange. Für jeden **Server** kann dabei z.B. festgelegt werden, wie viele Aktivitäten parallel abgearbeitet werden können und nach welcher Strategie Aktivitäten aus der Warteschlange entnommen werden. Der **Counter** dient als einfache Variable, die wahlweise erhöht oder erniedrigt werden kann. Welche Funktionalitäten eine Ressource zur Verfügung stellt, wird jeweils grafisch im Modell angezeigt. Für den **Server** ist dies **request** und für den **Counter** **change**. In Abbildung 1.2 ist eine weiter Ressource **Storage** abgebildet, die eine Erweiterung des **Counters** darstellt und weitere Funktionen zur Verfügung stellt.

Damit komplexe Systeme übersichtlich modelliert werden können, unterstützt das ProC/B-Paradigma hierarchische Strukturen. Teile einer Prozesskette können zu Funktionseinheiten zusammengefasst und wie normale Ressourcen von den Prozesskettenelementen einer höheren Ebene genutzt werden. Eine detaillierte Beschreibung dieser Modellierungsmöglichkeit ist in [BBK09] zu finden. Wie bei Warteschlangennetzen zielt eine Optimierung von ProC/B-Modellen darauf ab, gewisse Leistungsgrößen zu maximieren oder minimieren, indem die Parameter der Komponenten wie z.B. die Bedienraten der **Server** oder die Werte der **Counter** modifiziert werden.

### 1.1.3. Weitere Modelle

Neben den beiden bisher vorgestellten Formalismen existieren viele zum Teil kommerzielle Werkzeuge, die die Erstellung und Analyse von ereignisdiskreten Modellen unterstützen. Zu diesen gehören Arena [Roc] zur Modellierung von Fertigungssystemen, OMNeT++[Var] zur Modellierung von Computernetzen, die APNN-Toolbox [BFKT01] auf Basis stochastischer Petri-Netze und die Java Modelling Tools [dM06]

(JMT) auf Basis von Warteschlangenetzen. Diese Programme stellen darüber hinaus eine Reihe von Analyseverfahren zur Verfügung, die im Fall von OMNeT++, der APNN-Toolbox und JMT in OPEDo eingebunden sind.

## 1.2. Einsatzgebiete

In früheren Abschnitten wurde bereits motiviert, wieso Modelle von Systemen erstellt werden, und in welchen Situationen dies sinnvoll ist. Bevor auf eine allgemeinere Betrachtung von Modellen eingegangen wird, folgt eine kurze Übersicht von Fragestellungen, die typisch für den Einsatz von Modellen sind. Dabei existiert eine Reihe von Standardfragen, die in verschiedenen Variationen häufig auftreten:

- Wie hoch ist die Auslastung einzelner Maschinen?
- Wie häufig fallen bestimmte Komponenten aus?
- Wie lange dauert es, ein bestimmtes Produkt herzustellen?
- Welche Menge eines Produkts kann in einem Zeitintervall hergestellt werden?
- Wie hoch sind die Kosten eines Systems?
- Welches von zwei Systemen führt zu einem größeren Gewinn?

Die ersten vier Fragestellungen sind dabei rein technisch und können direkt durch eine Analyse eines Modells beantwortet werden. Im fünften Punkt ist eine wirtschaftliche Fragestellung beschrieben, für deren Beantwortung zusätzlich zu den technischen und im Modell feststellbaren Größen Informationen über entstehende Kosten berücksichtigt werden müssen. Beim letzten Punkt steht der Vergleich zweier Systeme im Vordergrund. Diese Fragestellung wird im Bereich der Optimierung im Fokus stehen.

## 1.3. Ereignisdiskrete Modelle

Die bisher aufgeführten Modelle können als ereignisdiskrete Modelle [LK99] interpretiert werden. Dabei wird davon ausgegangen, dass sich der Zustand eines Systems $S$ zu einem Zeitpunkt $t$ mit einer Menge $Z_t$ von Zustandsvariablen und dem Zustand der Ereignisliste, die zukünftige aber bereits geplante Ereignisse enthält, beschreiben lässt. Eine Veränderung des Zustands tritt nur ein, wenn Ereignisse eintreffen. Aus der Art des Ereignisses und dem bisherigen Zustand folgt der nächste Zustand. Die Zustandsänderung ist atomar. Für zwei Ereignisse $e_1$ und $e_2$ und deren Zeitpunkt $t_1$ und $t_2$, zwischen denen keine weiteren Ereignisse eingetreten sind, gilt also:

$$\forall t_a, t_b \in (t_1; t_2) : Z_{t_a} = Z_{t_b} \tag{1.1}$$

Diese Art der Zustandsänderung wird als diskret bezeichnet, da sie zu einem Zeitpunkt und nicht kontinuierlich über einen Zeitraum stattfindet. Daraus resultiert die Bezeichnung **ereignisdiskret**. Davon unberührt stammen die Zeitpunkte, zu denen die Ereignisse stattfinden, aus einem kontinuierlichen Wertebereich, der im Allgemeinen eine Teilmenge von $\mathbb{R}^+$ ist. Dabei ist zu beachten, dass zu einem Zeitpunkt $t$ mehrere Ereignisse eintreten können, so dass sich der Zustand zu einem Zeitpunkt mehrfach ändern kann. Die Reihenfolge der Änderungen ist in diesen Fällen teilweise nicht festgelegt.

Bei der Behandlung von Ereignissen kann nicht nur der Zustand geändert werden. Ereignisse können weitere Ereignisse zur Folge haben, wobei die direkten Nachfolger bei der Behandlung des Ereignisses geplant werden müssen. Hierbei wird festgelegt, welche Art von Ereignis zu welchem Zeitpunkt stattfinden soll. Ein typisches Beispiel ist das Eintreffen eines Kunden an einem Bedienschalter. Als Ereignis wird direkt das Bedienende des Kunden an diesem Schalter geplant. Anhand der Abfolge der Zustände kann das Verhalten einzelner Teile beobachtet und die erwünschten Messwerte erhoben werden.

Da ereignisdiskrete Modelle in der Regel stochastische Komponenten beinhalten, muss zu ihrer Analyse auf Methoden der Wahrscheinlichkeitsrechnung zurückgegriffen werden. Bei der folgenden Darstellung beschränke ich mich auf die zentralen Aspekte, die für die spätere Betrachtung notwendig sind. Detaillierte Einführungen in dieses Thema finden sich an verschiedenen Stellen in der Literatur. Die folgende Beschreibung orientiert sich an [Bau74] und verwendet die dort eingeführten Schreibweisen.

Zur Analyse werden ereignisdiskrete Modelle auf stochastische Prozesse abgebildet. Ein stochastischer Prozess besteht aus einer Menge von Zufallsvariablen $X_i$ und einem Messraum $(\Omega, \mathfrak{A})$, die den Zustand des Modells beschreiben. Dabei beschreibt $X_i$ den Zustand, der durch Eintreten eines Ereignisses zum Zeitpunkt $t_i$ erreicht wurde. Zur Beschreibung der Zustände des Modells sind $X_i$ in der Regel Vektoren, wobei $X_{i,j}$ die $j$-te Zustandsvariable des Modells beschreibt. Analog zu [Bau74] wird in dieser Arbeit die folgende Schreibweise genutzt:

$$(X_1, \ldots, X_n, t_1, \ldots, t_n) \tag{1.2}$$

Der diesen Modellen zugrunde liegende stochastische Prozess ist gekennzeichnet durch die Zeitpunkte $t_1, \ldots, t_n \in \mathbb{R}$ mit $t_i < t_{i+1}$ für $i = 1, \ldots, n-1$ an denen sich der Zustand des Modells ändern kann und den Zustandsvariablen $X_1, \ldots, X_n$ die den Zustand des Systems nach Eintreten der jeweiligen Ereignisse darstellen. Sowohl $t_i$ als auch $X_i$ sind dabei Zufallsvariablen.

## 1.4. Analyse ereignisdiskreter Modelle

Abhängig von den Analysezielen und der Struktur des stochastischen Prozesses existieren unterschiedliche Verfahren zur Analyse. Zu diesen gehören Simulationen, numerische Analysen von Markovketten und approximative Verfahren. Diese Verfahren unterscheiden sich vor allem in der Genauigkeit und der Art der Ergebnisse sowie in der benötigten Rechenzeit und sollen nun etwas genauer betrachtet werden.

### 1.4.1. Simulation

Unter der Simulation eines ereignisdiskreten Modells versteht man die schrittweise Abarbeitung der Ereignisse dieses Modells. Einführungen in dieses Thema finden sich unter Anderem in [LK99], [CL99] sowie [BNN05]. Darüber hinaus existieren praxisorientierte Beschreibungen zum Umgang mit Simulation und deren Ergebnissen wie zum Beispiel [Kle07c], [DK02] und [KD06]. Die Simulation stellt dabei die einzige Analysemethode dar, mit der ereignisdiskrete Modelle in allgemeiner Form interpretiert werden können.

Bei der Betrachtung von Modellen mittels Simulation muss zwischen der transienten und stationären Analyse unterschieden werden. Diese beiden Herangehensweisen unterscheiden sich in der Art des Abbruchkriteriums. Dabei wird bei der transienten Analyse der Zeitpunkt, zu dem die Simulation beendet wird, vor Beginn der Simulation festgelegt und die Messwerte bei Beendigung der Simulation bestimmt. Diese Art der Analyse ist z.B. bei Modellen interessant, die zu bestimmten Zeitpunkten wieder in ihren Ursprungszustand versetzt werden, wie es z.B. durch eine regelmäßige Reinigung aller beteiligten Maschinen der Fall sein kann. Um zusätzliche Trajektorien zu erzeugen, können bei der transienten Analyse die Simulationen mit unterschiedlichen Startwerten des Zufallszahlengenerators wiederholt werden, wodurch alternative Verläufe über die Simulation durchschritten werden.

Bei der stationären Analyse soll ermittelt werden, wie sich das Modell bzw. die zu messenden Werte verhalten, wenn das Modell unendlich lange ausgeführt wird. Zwar werden Systeme nicht gebaut, um ewig eingesetzt zu werden, jedoch nähert sich das Verhalten eines Systems im Dauerbetrieb den so ermittelten Werten an. Ein Beispiel für eine solche Situation ist eine Fabrik im Dreischichtbetrieb. Voraussetzung für diese Analyse ist jedoch, dass sich die zu messenden Werte bei der Analyse einem Wert annähern. Dies bedeutet, dass der stochastische Prozess einen stationären Zustand erreicht. Sei $Y$ die zu untersuchende Zufallsvariable und $Y(t)$ die Zufallsvariable zum Zeitpunkt $t$, soll $lim_{t\to\infty}Y(t) = Y$ existieren. Da jedoch nur Erwartungswerte betrachtet werden, genügt die Existenz von $lim_{t\to\infty}E[Y(t)] = E[Y]$. Die Erwartungswerte der Zufallsvariablen nähern sich dementsprechend mit steigender Zeit dem zu ermittelnden Erwartungswert an. Wenn ein stationärer Zustand existiert, gibt es ein $t_s$ mit $E[Y(t)] \approx E[Y]$ für $t \geq t_s$. Die Simulation wird so in eine Aufwärmphase (auch transiente Phase genannt) und in die stationäre Phase unterteilt. Für die Schätzung

von $E[Y]$ wird bei bekanntem $t_s$ nur der Teil der ermittelten Stichprobe verwendet, der in der stationären Phase erhoben wurde. Die Bestimmung von $t_s$ ist jedoch nicht trivial. Grundsätzlich müssen dabei zwei Positionen gegeneinander abgewogen werden: Zum einen sollte $t_s$ möglichst groß gewählt werden, um sicherzustellen, dass die stationäre Phase erreicht wurde. Zum anderen sollte $t_s$ möglichst klein gewählt werden, um den Rechenaufwand gering zu halten. Zur Bestimmung von $t_s$ existieren verschiedene Verfahren wie z.B. die Methoden von Welch [Wel81], Gallagher, Bauer und Maybeck [GKBM96], die hier jedoch nicht beschrieben werden. Eine Übersicht dieser und weiterer Verfahren ist in [LK99] zu finden.

Wie bereits in Abschnitt 1.3 beschrieben, enthalten ereignisdiskrete Modelle im Allgemeinen Zufallsvariablen. Daraus resultiert, dass alle Messwerte ebenfalls als Zufallsvariablen angesehen werden müssen, wobei die Verteilung dieser Zufallsvariablen nicht näher bekannt ist. Da die Simulation eines Modells lediglich eine Ausführung des Modells mit jeweils erzeugten Realisierungen der im Modell festgelegten Zufallsvariablen ist, ist das Ergebnis lediglich eine Stichprobe und damit ggf. unbekannten Schwankungen unterworfen. Daher sind in der Regel zusätzliche statistische Untersuchungen der Stichprobe notwendig.

Bei der bisherigen Beschreibung wurde nicht berücksichtigt, dass seltene Ereignisse wie z.B. Ausfälle von Komponenten oder Pufferüberläufen mittels Simulation nur schwierig zu erfassen sind. Diese müssen bei der Wahl dieser Analysemethode gesondert berücksichtigt werden.

In der Praxis sind ereignisdiskrete Simulationsmodelle sehr beliebt, da sie wenigen Einschränkungen unterliegen und häufig intuitiv benutzbar sind. Die große Auswahl an Simulatoren und Modellierungswerkzeugen spiegelt dies wieder. Eines der bekanntesten Programme ist Arena [Roc], im Open-Source-Bereich wird häufig auf OMNeT++ [Var] gesetzt, das auch im wissenschaftlichen Bereich auf steigendes Interesse stößt. Eine speziell für die Logistik zugeschnittene Software wurde im Rahmen des SFB559 [BBK09] entwickelt. Diese steht unter dem Namen ProC/B-Toolset [BBF+02] zur Verfügung, das die Modellierung mit Prozessketten umsetzt, wie sie im ProC/B Paradigma beschrieben sind [BBS03, BBK98]. Neben dem bisher genutzten Simulator HIT, der am Lehrstuhl 4 der TU Dortmund entwickelt wurde, wird in aktuellen Versionen auch OMNeT++ unterstützt [KV08].

### 1.4.2. Numerische Analyse von Markov-Ketten

Eine weitere Möglichkeit, eine eingeschränkte Klasse von ereignisdiskreten Modellen zu analysieren, ist die numerische Analyse von Markov-Ketten. Bei dieser Methode wird zuerst der dem Modell zugrunde liegende stochastische Prozess vereinfacht, da Markov-Ketten [Mar06, Mar71] eine Klasse von stochastischen Prozessen mit bestimmten Einschränkungen sind. An dieser Stelle sind nur die zeitkontinuierlichen Markov-Ketten (engl. continuous-time Markov chain (CTMC)) von Interesse, da

diese nur eine geringe Vereinfachungen der stochastischen Prozesse benötigen. Die hier folgende Beschreibung orientiert sich an [Ste94].

Ein stochastischer Prozess $\{X(t), t \geq 0\}$ mit einem Zustand $X(t) \in \{X_1, \ldots, X_l\}$ zum Zeitpunkt $t$ und $l$ Zuständen ist eine CTMC, wenn für alle $n \in \mathbb{N}$ und für jede Sequenz $t_0 < t_1 < \ldots < t_n < t_{n+1}$ gilt:

$$\begin{aligned} Prob\{X(t_{n+1}) = x_{n+1} | X(t_0) = x_0, X(t_1) = x_1, \ldots, X(t_n) = x_n\} = \\ Prob\{X(t_{n+1}) = x_{n+1} | X(t_n) = x_n\} \end{aligned} \tag{1.3}$$

Wenn sich die CTMC zum Zeitpunkt $s$ im Zustand $X_i$ befindet, ist die Wahrscheinlichkeit zum Zeitpunkt $t$ im Zustand $X_j$ zu sein, definiert als:

$$p_{ij}(s,t) = Prob\{X(t) = j | X(s) = X_i\} \tag{1.4}$$

Diese Schreibweise wird im Allgemeinen nur für inhomogene CMTCs benutzt, bei denen sich die Zustandswechselwahrscheinlichkeit über die Zeit ändern kann. Bei homogenen CMTCs bleibt die Zustandswechselwahrscheinlichkeit über die Zeit gleich, daher gilt für homogene CTMCs darüber hinaus für beliebige $t \geq s$ mit $\tau = t - s$:

$$p_{ij}(\tau) = Prob\{X(s+\tau) = j | X(s) = X_i\} = p_{ij}(s,t) \tag{1.5}$$

Des Weiteren gilt für die Wechselwahrscheinlichkeiten, dass deren Summe gleich 1 ist:

$$\forall i \in \{1, \ldots, l\} : \sum_{j=1}^{l} p_{ij}(\tau) = 1 \tag{1.6}$$

Im weiteren Verlauf werden ausschließlich homogene CTMCs betrachtet. Die Matrix der Zustandsübergangswahrscheinlichkeiten eines homogenen CTMC ist $P(\tau) = (p_{ij}(\tau))$. Die Übergangsraten von einem Zustand $X_i$ in einen Zustand $X_j$ können wie folgt berechnet werden:

$$q_{ij} = \lim_{\Delta t \to 0} \left\{ \frac{p_{ij}(\Delta t)}{\Delta t} \right\} \quad \text{mit} \quad i \neq j \quad \text{und} \quad q_{ii} = -\sum_{j:i \neq j} q_{ij} \tag{1.7}$$

Dadurch erhält man die Generatormatrix $Q = (q_{ij})$, die auch Transitionsratenmatrix genannt wird. Wenn $\pi_t$ mit $\pi_{t,i} = Prob\{X(t) = X_i\}$ die Wahrscheinlichkeit für Zustand $X_i$ zum Zeitpunkt $t$ bei bekannter Verteilung $\pi_0$ zum Zeitpunkt 0 ist, gilt folgender Zusammenhang:

$$\frac{\partial \pi_t}{\partial t} = \pi_t \cdot Q \tag{1.8}$$

Daraus kann folgende Formel für $\pi_t$ abgeleitet werden:

$$\pi_t = \pi_0 e^{Qt} = \pi_0 \sum_{k=0}^{\infty} \frac{(Qt)^k}{k!} \tag{1.9}$$

Diese Beziehung kann zur transienten Analyse (siehe Kapitel 1.4.1) von homogenen CTMCs genutzt werden. Weitere Algorithmen zur transienten Analyse von Markov-Prozessen sind in [Ste94] zu finden. Darüber hinaus können daraus die Zusammenhänge für die stationäre Analyse (siehe Kapitel 1.4.1) abgeleitet werden. Voraussetzung für die stationäre Analyse ist die Existenz eines Zeitpunkts $t$ für den gilt:

$$\forall s > 0 : \pi_t = \pi_{t+s} \tag{1.10}$$

Damit gilt für die Verteilung $\pi_t$:

$$\partial \pi_t / \partial t = 0 \tag{1.11}$$

Die Verteilung $\pi_t$ heißt dann stationäre Verteilung $\pi$. Aus Gleichung 1.8 ergibt sich:

$$\pi \cdot Q = 0 \tag{1.12}$$

Zur Berechnung von $\pi$ existieren verschiedene Verfahren, die an dieser Stelle nicht im Detail betrachtet werden sollen. Grundsätzlich unterteilen sich diese in direkte und iterative Verfahren. Direkte Verfahren haben jedoch abhängig von der Anzahl $l$ der Zustände eine Laufzeit von $O(l^3)$ und einen Platzbedarf von $O(l^2)$, womit die Rechenzeit und der Platzbedarf für große Markov-Ketten stark ansteigt. Einen Ausweg bieten iterative Verfahren. Diese nähern sich ausgehend von einer Anfangsverteilung $\pi$ an, wobei die Anzahl der Berechnungen, die diese Verfahren benötigen, um $\pi$ bis zu einer vorher festzulegenden Genauigkeit zu berechnen, von der Wahl des Anfangsvektors $\pi_0$ abhängt. Darüber hinaus hat die Genauigkeit der Berechnung einen erheblichen Einfluss auf die Rechenzeit und sollte daher so gewählt werden, dass die Ergebnisse ausreichend verlässlich sind und die Rechenzeit nicht unverhältnismäßig hoch ist. Die Wahl dieser Werte hängt daher insbesondere vom zu Grunde liegenden Modell und den weiteren Berechnungen ab. Daher können diese Werte nicht allgemein festgelegt werden.

Nachdem der Verteilungsvektor $\pi$ bestimmt wurde, können seine Informationen mit der Beschreibung des Zustandsraums, der vom ursprünglichen Modell abhängt, kombiniert werden. Dadurch können die Leistungsgrößen für das zu Grunde liegende Modell berechnet werden.

### 1.4.3. Approximative Verfahren

Der Bereich der approximativen Verfahren ist sehr weitläufig. Da er in dieser Arbeit nicht weiter betrachtet wird, wird hier nur kurz ein Hinweis auf den grundlegenden Ansatz sowie weiterführende Literatur gegeben. Im Allgemeinen werden hier Verfahren zusammengefasst, die eine direkte und einfache Berechnung eines Näherungswerts für ein bestimmtes und in der Regel relativ kleines Modell ermöglichen, wobei ggf. die Ergebnisse für kleine Teilmodelle kombiniert werden können und so zu einer approximativen Lösung des Gesamtmodells führen. Es handelt sich dementsprechend um Verfahren, deren Einsatzfähigkeit auf ein bestimmtes Gebiet beschränkt ist. Zum Beispiel existieren approximative Verfahren für erweiterte Fork/Join-Warteschlangennetze [Arn06] und für spezielle geschlossene Warteschlangennetze [uYD93, Hav95]. Eine Übersicht über diesen Themenkomplex und weiterführende Literatur findet sich in [Arn06]. Einige dieser Verfahren sind in den Java Modelling Tools [dM06] umgesetzt.

# 2. Ereignisdiskrete Modelle und Funktionen

Ereignisdiskrete Modelle können i.d.R. leicht modifiziert werden. So können Untersuchungen an unterschiedlichen Varianten eines Modells durchgeführt werden. Dabei können zwei unterschiedliche Typen von Veränderungen unterschieden werden. Bei der ersten wird die Struktur des Modells geändert. Es werden z.B. alternative Fertigungsstraßen im Modell einer Fabrik eingeführt. Bei der zweiten Variante werden ausschließlich qualitative oder quantitative Aspekte eines Modells modifiziert, die Struktur jedoch beibehalten. Dies trifft z.B. auf veränderte Bedienraten bei Servern, geänderte Puffergrößen oder alternative Bearbeitungsstrategien zu. In diesem zweiten Fall sprechen wir von Modellen mit Freiheitsgraden bzw. Parametern. Diese Parameter können mit unterschiedlichen Werten belegt werden. Zu beachten ist hierbei, dass in gewissen Grenzen Strukturänderungen in Parametern kodiert werden können. So kann z.B. zwischen zwei Fertigungsstraßen über eine Weiche im Modell umgeschaltet werden, die über einen boolschen Parameter gesteuert wird. In diesen Fällen ist es jedoch sinnvoller, zwei alternative Modelle zu betrachten und die Ergebnisse anschließend zu vergleichen. Daher werden solche Strukturänderungen im Weiteren nicht mehr betrachtet. Die folgende Beschreibung orientiert sich an [LK99].

Durch Modelle mit Freiheitsgraden können verschiedene zusätzliche Fragestellungen bearbeitet werden. Typische Fragestellungen sind:

1. Bei welcher Bedienrate erreicht eine Maschine ihr bestes Kosten/Nutzen-Verhältnis?

2. Wie groß muss ein Lager sein, damit keine Engpässe bestimmter Ressourcen auftreten?

3. Wie viele Maschinen mit gleicher Geschwindigkeit müssen parallel eingesetzt werden, um einen maximalen Gewinn zu erwirtschaften?

Dazu werden die Parameter mit Werten belegt (Eine Belegung aller Parameter wird **Konfiguration** genannt). Indem systematisch unterschiedliche Konfigurationen analysiert und verglichen werden, können diese Fragestellungen beantwortet werden (siehe Kapitel 5).

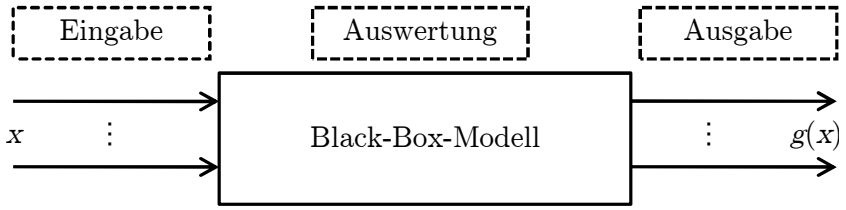



Abbildung 2.1.: Black-Box-Modell: Bei der Betrachtung des Modells werden nur die zu den jeweiligen Eingaben erhaltenen Ausgaben berücksichtigt.

**Definition 1 (Modellkonfiguration)** *Eine **Modellkonfiguration** $x$ entspricht einer Belegung der freien Parameter eines Modells. Ein Modell $M$ mit der Modellkonfiguration $x$ wird als $M_x$ geschrieben.*

Da im Allgemeinen zu Funktionen, die auf ereignisdiskreten Modellen basieren, keine Ableitungen oder weitere Informationen bestimmt werden können, werden im Folgenden ereignisdiskrete Modelle als Black-Box-Modelle betrachtet. Bei dieser Sichtweise wird das Modell selbst nicht detaillierter betrachtet und darauf reduziert, dass eine Menge von Parametern $x$, die den möglichen Modellkonfigurationen entsprechen, zu einer Menge von Ausgaben oder Messwerten $g(x)$ führt (siehe Abbildung 2.1).

**Definition 2 (Black-Box-Modell)** *Ein **Black-Box-Modell** $M$ ist ein Modell mit Parametern, an dem eine Reihe von Werten gemessen werden kann. Zur Struktur dieser Modelle stehen keine weiteren Informationen zur Verfügung.*

Aus formaler Sicht ist eine Auswertung eines Black-Box-Modells $M$ die Bestimmung des Werts einer Funktion $g_M$ für die Modellkonfiguration $x$, die einen Vektor $c = (c_1, \ldots, c_l)'$ der $l$ gemessenen Größen bestimmt. Den Eingabevektor $x$ wird im Folgenden im Umfeld der Funktionen als Punkt bezeichnet. Die Zielfunktion $f_M(x)$ kann daher geschrieben werden als:

$$f_M(x) = h_M(g_M(x)) \tag{2.1}$$

wobei $h_M$ die Zielfunktion mittels der mit $g_M$ gemessenen Werte bestimmt.

Bei ereignisdiskreten Modellen ist $f_M(x)$ in der Regel eine Zufallsvariable, da die Werte des Vektors $c$ aufgrund der stochastischen Eigenschaften der Modelle ebenfalls Zufallsvariablen sind. Abhängig vom Analyseverfahren beschreibt $f_M(x)$ daher ggf. kein eindeutiges Ergebnis und kann bei mehreren Auswertungen unterschiedliche Werte liefern. Daher wird versucht, den Erwartungswert $E[f_M(x)]$ sowie die Varianz $\sigma^2(f_M(x))$ zu bestimmen oder zu approximieren. Im Allgemeinen kann man nicht davon ausgehen, diese beiden Werte korrekt berechnen zu können. Stattdessen

werden sie ausgehend von einer Stichprobe geschätzt. Die Schätzer für den Erwartungswert $\hat{y}$ und die Varianz $\hat{\sigma}^2$ für eine Stichprobe $y_i, \ldots, y_n$ bei $f_M(x)$ sind, sofern keine weiteren Informationen über $f_M(x)$ vorliegen, durch die folgenden Formeln gegeben:

$$\hat{y} = \frac{1}{n} \sum_{i=1}^{n} y_i \qquad \hat{\sigma}^2 = \frac{1}{n-1} \sum_{i=1}^{n} (y_i - \hat{y})^2 \tag{2.2}$$

Bei ausreichend großen Stichproben konvergiert die Wahrscheinlichkeit, dass die Schätzer um mehr als $\varepsilon > 0$ vom wahren Wert abweichen gegen 0. Darüber hinaus kann nicht davon ausgegangen werden, dass die Varianz für alle möglichen Konfigurationen identisch ist. In vielen Fällen kann man jedoch annehmen, dass die Varianz bei zwei nahe beieinander liegenden Punkten ähnlich ist. Die Varianzen der zugehörigen Stichproben können sich trotzdem signifikant unterscheiden.

Zusätzlich zu den Informationen, die man durch die Schätzer für den Erwartungswert und die Varianz gewonnen hat, ist in der Regel eine Aussage darüber interessant, in welchem Bereich der Erwartungswert wahrscheinlich liegt. Diese Aussage wird über Konfidenzintervalle getroffen. Dabei gibt ein $\beta$-Konfidenzintervall an, dass der Erwartungswert mit einer Wahrscheinlich von mindestens $\beta = 1 - \alpha$ in diesem Intervall enthalten ist. Die Größe des Konfidenzintervalls hängt zum Einen von der Varianz der Stichprobe und zum Anderen von den Informationen über die der Stichprobe zugrunde liegenden Verteilung ab. Unter den Annahmen, dass die Varianz aus der Stichprobe geschätzt werden muss und die Stichprobe normalverteilt ist, ergibt sich folgende Formel für die Berechnung des Konfidenzintervalls:

$$\left[ \hat{y} - t_{1-\frac{\alpha}{2},n-1} \frac{\hat{\sigma}}{\sqrt{n}}; \hat{y} + t_{1-\frac{\alpha}{2},n-1} \frac{\hat{\sigma}}{\sqrt{n}} \right] \tag{2.3}$$

wobei $t_{\alpha,n}$ das $\alpha$-Quantil der t-Verteilung mit $n$ Freiheitsgraden ist.

Um die Stichproben zu mehreren Modellkonfigurationen zu vergleichen, existieren verschiedene Verfahren wie z.B. das Paired-t-Verfahren [LK99]. Diese Verfahren beurteilen, ob sich die Stichproben und die zugehörigen Konfidenzintervalle ausreichend gut unterscheiden lassen.

Abschließend sei darauf hingewiesen, dass zu einem Modell mehrere Funktionen betrachtet und mittels einer Modellauswertung berechnet werden können. Im weiteren Verlauf dieser Arbeit ist die Betrachtung ausschließlich auf eine Funktion zu einem Modell begrenzt. Die Bewertung eines Modells erfolgt dementsprechend ausschließlich anhand eines Kriteriums.

## 2.1. Einschränkungen der betrachteten Funktionen

Da in dieser Arbeit ereignisdiskrete Modelle als Black-Boxes betrachtet werden, existieren keine genaueren Kenntnisse über das „Innere“ eines Modells bzw. zu der durch das Modell dargestellten Funktion. Insbesondere können im Allgemeinen zu der Funktion keine Ableitungen oder Integrale bestimmt werden und der Verlauf der Funktion ist unbekannt. Dadurch ist auch nicht bekannt, ob eine Funktion innerhalb des Wertebereichs $W$ stetig oder nicht stetig, uni- oder multimodal ist. Die Werte der Funktion können jedoch punktweise geschätzt bzw. berechnet werden. Zur Untersuchung der Funktion können dementsprechend nur die Funktionswerte für unterschiedliche Konfigurationen verglichen und dadurch Rückschlüsse auf einen wahrscheinlichen Verlauf der Funktion gezogen werden.

Es ist nicht sinnvoll, beliebige Funktionen zu betrachten. Zur Veranschaulichung dieser Aussage soll eine Funktion $f_a(x)$ mit einem konstanten Vektor $a \in W$ und einem Wertebereich $W \subset \mathbb{R}^n$ dienen, deren Funktionswert an der Stelle $x = a$ gleich 0 und an allen anderen Stellen 1 ist. Da diese Funktion keine Struktur aufweist, die auf das Minimum bei $a$ hinweist, ist eine systematische Annäherung an dieses Minimum nicht möglich. Ein Erreichen dieses Minimums kann nicht garantiert werden.

Die durch ereignisdiskrete Modelle gegebenen Funktionen weisen jedoch in der Regel gewisse Strukturen auf, z.B. gilt für die meisten nahe beieinander liegenden Punkte, dass auch deren Funktionswerte ähnlich sind. Auf Modelle dieser Art beschränkt sich diese Arbeit daher im Folgenden. Um diese Strukturen genauer beschreiben zu können, müssen der Mittelpunkt einer konvexen Menge sowie der kleinste Durchmesser definiert werden.

**Definition 3 (Mittelpunkt einer konvexen Menge)** *Ein Punkt $x$ ist Mittelpunkt einer konvexen Menge $M$, wenn es keinen Punkt $y \in M$ gibt, dessen durchschnittliche Distanz zu allen anderen Punkten in $M$ kleiner als die des Punkts $x$ ist.*

**Definition 4 (Kleinster Durchmesser)** *Sei $W \subset \mathbb{R}^n$ eine konvexe Menge, $G$ die Menge aller Geraden im $\mathbb{R}^n$, die den Mittelpunkt von $W$ enthalten, und $S$ die Menge aller Strecken, die dadurch entsteht, dass alle Geraden aus $G$ mit $W$ geschnitten werden. Die Länge der kürzesten Strecke in $S$ ist dann der kleinste Durchmesser von $W$. Darüber hinaus sei der kleinste Durchmesser einer Menge von konvexen Mengen $M$ das Minimum der kleinsten Durchmesser aller Elemente aus $M$.*

Mit diesen Definitionen können die Funktionen genauer festgelegt werden, auf die sich diese Arbeit beschränkt. Eine deterministische Funktion $f(\mathbf{x})$ mit Wertebereich $\mathbf{W}$, einer Teilmenge von $\mathbb{R}^n$, darf Sprungstellen enthalten. Es muss eine Menge von Partitionen $\mathbf{W}_T$ existieren, die folgende Eigenschaften erfüllt:

1. $\mathbf{W}_T = \{\mathbf{W}_1, \ldots, \mathbf{W}_m\}$ ist eine Partition von $\mathbf{W}$.

2. Alle $\mathbf{W}_i \in \mathbf{W}_T$ sind konvex.

3. $\forall i \in \{1, \ldots, m\} \wedge \forall \varepsilon_2 > 0 : \forall x \in \mathbf{W}_i : \exists \varepsilon_1 > 0 :$
   $\forall x' \in W_i \wedge |x' - x| \leq \varepsilon_1 : |f(x) - f(x')| \leq \varepsilon_2$.

4. Für den kleinsten Durchmesser $d_{min}$ (siehe Definition 4) aller $W_i$ gilt, dass es einen kleinen Faktor $l \in \mathbb{R}^+$ gibt, so dass $d_{min} * l$ nicht kleiner als der kleinste Durchmesser von $W$ ist.

5. Die Menge $W_T$ ist nicht bekannt.

Die Punkte 1 bis 2 stellen sicher, dass die Mengen $\mathbf{W}_i$ den gesamten Wertebereich $\mathbf{W}$ abdecken, sich nicht überschneiden und jeweils zusammenhängend sind. Punkt 3 beschreibt, dass es für jeden Punkt $x$ einen Umkreis gibt, so dass sich die Funktionswerte der Punkte im Umkreis nicht mehr als ein Epsilon von dem Funktionswert von $x$ unterscheiden. Die Funktion $f$ muss innerhalb beliebiger $\mathbf{W}_i$ nicht unimodal sein. Punkt 4 legt fest, dass alle $\mathbf{W}_i$ im Verhältnis zu $\mathbf{W}$ ausreichend groß sein müssen. Punkt 5 stellt grundsätzlich keine Einschränkung dar, jedoch ist es bei Kenntnis von $\mathbf{W}_T$ i.d.R. sinnvoller, die Funktion für die einzelnen Teilmengen separat zu betrachten. Da bei der Simulation ereignisdiskreter Modelle in der Regel keine deterministischen Funktionen entstehen, müssen in diesem Fall die oben genannten Bedingungen für den Erwartungswert von $f$ erfüllt und die Varianz endlich sein.

## 2.2. Benchmarks

In dieser Arbeit werden verschiedene empirische Untersuchungen durchgeführt. Die Basis für viele Testreihen werden die im folgenden Abschnitt beschriebenen Benchmarkmodelle und -funktionen sein. Zu diesem Zweck werden die Modelle einer Produktionslinie und eines Lagerhaltungssystems herangezogen, die in der Literatur häufig genutzt werden. Da die Analyse von ereignisdiskreten Modellen sehr zeitaufwändig ist, werden für einige erweiterte Testreihen Benchmarkfunktionen eingesetzt.

### 2.2.1. Produktionslinie

Das System der Produktionslinie besteht aus $l$ Maschinen, die in Reihe geschaltet sind. Teile erreichen die erste Maschine $M_1$, werden dort bearbeitet und nach einer Bearbeitungsdauer an die zweite Maschine $M_2$ weitergeleitet. Dies setzt sich an den folgenden Maschinen fort. An der letzten Maschine werden die Teile weiterverarbeitet und verlassen das System. Die Bedienrate der Maschinen ist exponentialverteilt. Die mittlere Bedienrate sei $b_i$ für $M_i$ mit $i = 1, \ldots, l$. Die Bedienraten werden im Vektor $b = (b_i, \ldots, b_l)$ zusammengefasst. Zusätzlich können vor jeder Maschine $K = 10$ Teile warten, überschüssige Teile verlassen das System ohne weiter verarbeitet zu

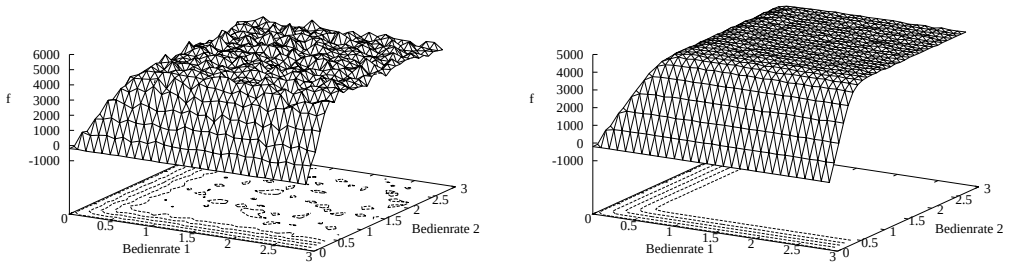



Abbildung 2.2.: Darstellung der Funktion $f(b)$ der Produktionslinie für zwei Maschinen (links ohne Replikationen, rechts mit 100 Replikationen)

werden. Neue Teile erreichen die Maschine $M_1$ in exponentialverteilten Abständen mit $\lambda = 0,5$.

Für den Einsatz der Produktionslinie treten Fixkosten in Höhe von $c_0 = 200\$$ auf. Die Kosten für den Betrieb der Maschinen ist abhängig von deren Bedienrate und beträgt $c_i = d_i * b_i$ für $i = 1, \ldots, l$, wobei $d_i = i * 10\$$ die Kosten für den Betrieb der Maschinen sind, die mit der gewählten Bedienrate multipliziert werden, d.h. der Betrieb von Maschine $M_2$ kostet bei gleicher Bedienrate doppelt soviel wie der Betrieb von Maschine $M_1$. Teile, die die Produktionslinie erfolgreich durchlaufen haben können zu einem Preis $p = 100\$$ verkauft werden.

Durch ein Modell dieses Systems kann der Durchsatz von Maschine $M_l$ ermittelt werden. Da der Durchsatz abhängig von den Bedienraten $b$ ist, kann er als Funktion $g(b)$ betrachtet werden. Der durchschnittliche Gewinn je Zeiteinheit dieses Modells berechnet sich dann wie folgt:

$$f(b) = g(b) \cdot p - \sum_{i=1}^{l} (d_i b_i) - c_0 \tag{2.4}$$

Der Wertebereich für $b_i$ ist das Intervall $[0,001; 3]$. Der Verlauf der Funktion für $l = 2$ ist in Abbildung 2.2 dargestellt. Das Optimierungsproblem besteht darin, die Bedienraten $b$ zu bestimmen, für die $f(b)$ maximal ist. Die Simulationsdauer beträgt jeweils 100 Zeiteinheiten. Das bei 1000 Replikationen mittels Simulationsexperimenten auf zwei Nachkommastellen bestimmte globale Optimum für $l = 2$ ist $x_{max} = (0,66; 0,54)$. Die besondere Herausforderung dieses Optimierungsproblems liegt darin, dass im Bereich für $x_i$ größer als $0,7$ die Steigung der Funktion sehr gering ist. Diese kann, wie im linken Teil von Abbildung 2.2 zu sehen, bei zu geringen Anzahl von Replikationen durch Abweichungen in den Messwerten überlagert werden.

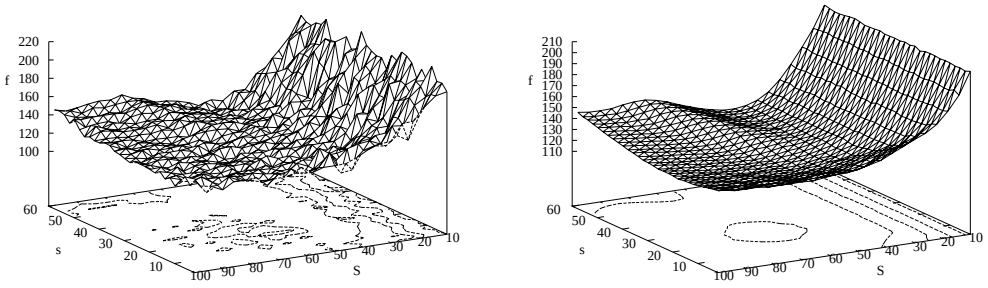



Abbildung 2.3.: Darstellung der monatlichen Kosten des Lagerhaltungssystems nach Law/Kelton [LK99] (links ohne Replikationen, rechts mit 100 Replikationen)

### 2.2.2. Lagerhaltungssystem

Fragestellungen für Lagerhaltssysteme gehören im Bereich der Betriebswirtschaftslehre zu den Standardproblemen, wodurch eine Vielzahl unterschiedlicher Beschreibungen entstanden ist. Das hier betrachtete System richtet sich nach der Beschreibung in [LK99].

Eine Firma verkauft genau eine Ware, die sie in einem Lager der Größe $S$ vorhält. Monatlich wird überprüft, ob der Lagerbestand $l$ unter eine Schwelle $s$ gefallen ist. Ist dies der Fall wird eine Bestellung bei einem Großhändler durchgeführt, bei der $S - l$ Teile der Ware bestellt werden. Dabei treten Bestellkosten $c_0 = 32\$$ und Stückkosten $c_1 = 3\$$ auf. Die Zeit, die zwischen der Bestellung und der Lieferung der Ware vergeht, ist gleichverteilt im Intervall [0,5; 1] Monat.

Kunden rufen zu unterschiedlichen Zeitpunkten bestimmte Mengen dieser Ware ab. Die Zwischenankunftszeiten der Bestellungen ist exponentialverteilt mit Mittelwert $0,1$ Monat. Die Menge $m$ der Ware, die ein Kunde abruft, ist wie folgt verteilt:

$$m = \begin{cases} 1 & ; \quad m.W. \ \frac{1}{6} \\ 2 & ; \quad m.W. \ \frac{1}{3} \\ 3 & ; \quad m.W. \ \frac{1}{3} \\ 4 & ; \quad m.W. \ \frac{1}{6} \end{cases} \quad (m.W. = mit\ Wahrscheinlichkeit) \tag{2.5}$$

In [LK99] werden ausschließlich die Kosten dieses Systems je Monat betrachtet. Die Kosten für eine Bestellung setzen sich aus den Bestellkosten in Höhe von $b_0 = 32\$$ und den Kosten je Einheit $b_1 = 3\$$ zusammen. Für jede Einheit der Ware im Lager treten für den entsprechenden Zeitraum Lagerhaltungskosten in Höhe von monatlich $c_0 = 1\$$ auf. Für jeden Kunden, der nicht bedient werden kann, weil nicht mehr genügend Einheiten im Lager vorhanden sind, werden Strafkosten in Höhe von $m \cdot 5\$$ berechnet.

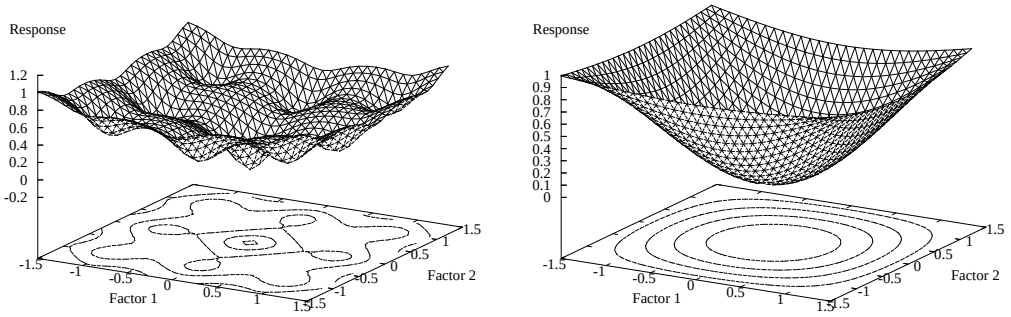



Abbildung 2.4.: Darstellung der Ackley- und der Sinus-Funktion für 2 Dimensionen

Die Bestellschwelle $s$ und die Lagergröße $S$ können in einem Modell dieses Systems modifiziert werden. Dabei gilt $s \in \{1, 2, \ldots, 60\}$ und $S \in \{10, 11, \ldots, 100\}$. Die Simulation wird jeweils für 10 Jahre durchgeführt. Der Verlauf der Funktion ist in Abbildung 2.3 dargestellt.

Das bei 5000 Replikationen ermittelte Optimum liegt bei $s = 24$ und $S = 62$. Die ermittelten Werte stimmen dabei mit den 9 unterschiedlichen Konfigurationen überein, die in [LK99] betrachtet wurden. Die dem ermittelten Optimum am nächsten liegende Konfiguration $s = 20$ und $S = 60$ wurde dabei als die beste der 9 Konfigurationen bestimmt.

### 2.2.3. Benchmarkfunktionen

In dieser Arbeit werden verschiedene Benchmarkfunktionen eingesetzt, um gezielt bestimmte Problemstellungen zu definieren. Diese Funktionen sind aus [NP98] und [Sal98] sowie [Poh94] entnommen. Die Auswahl enthält die Ackley- (Abbildung 2.4), Sinus- (Abbildung 2.4), Goldstein-Price- (Abbildung 2.5) und Schaffer-Funktion (Abbildung 2.5). Diese Funktionen decken einen großen Bereich unterschiedlicher Funktionen ab. Die Sinus-Funktion ist im gewählten Funktionsbereich unimodal. Die Ackley-Funktion ist multimodal mit vier lokalen Optima, deren Funktionswerte sich jedoch deutlich von dem des globalen Optimums unterscheiden. Die Schaffer-Funktion ist stark multimodal mit ringförmigen lokalen Optima, deren Werte nah an dem des globalen Optimums liegen. Die Goldstein-Price-Funktion ist unimodal und enthält im gewählten Wertebereich eine große Fläche, deren Funktionswerte sich nur geringfügig unterscheiden. Zu einigen Rändern sowie nahe dem globalen Optimum ist die Steigung jedoch deutlich größer; sowohl die Approximation der Randbereiche als auch die Approximation des Bereichs des globalen Optimums stellen eine Herausforderung dar. Mit diesen Funktionen können daher viele unterschiedliche Eigenschaften von Funktionen bei der Untersuchung der im weiteren Verlauf dieser Arbeit beschriebenen Verfahren berücksichtigt werden. Die Definitionen der vier

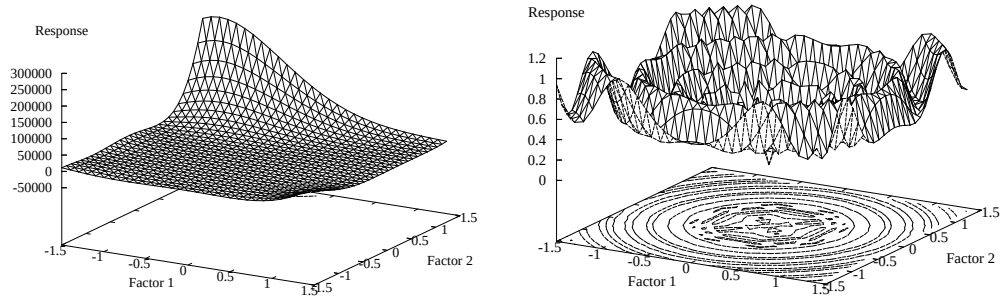



Abbildung 2.5.: Darstellung der Goldstein-Price- und der Schaffer-Funktion für 2 Dimensionen

Funktionen sind in Tabelle 2.1 angegeben.

Die Funktionen wurden zur einfacheren Vergleichbarkeit der Ergebnisse auf einen Wertebereich von ca. 0 bis 1 und die Werte der einzelnen Parameter auf ein Intervall von [-1,5; 1,5] skaliert. Das Optimum liegt bei diesen Funktionen mit Ausnahme der Goldstein-Price-Funktion im Nullpunkt; bei der Goldstein-Price-Funktion liegt es im Punkt (0, -1). Darüber hinaus sind die Goldstein-Price- und die Schaffer-Funktion im Allgemeinen für genau zwei Dimensionen definiert. Da die Benchmarkfunktionen für beliebig viele Dimensionen einsetzbar sein sollen, müssen diese entsprechend angepasst werden. Zu diesem Zweck werden bei gerader Anzahl von Dimensionen $k$ die Werte $x_i$ eines Punkts $x$ entsprechend ihrer Reihenfolge paarweise gruppiert, für jedes Paar der Wert der Funktion berechnet und die Ergebnisse summiert. Ist $k$ ungerade wird $x$ um eine Dimension erweitert und die Berechnung erneut begonnen. Damit ist $k$ gerade und der Funktionswert kann entsprechend berechnet werden. Der zusätzliche Wert beträgt dabei für die Goldstein-Price-Funktion $-1$ und für die Schaffer-Funktion 0. Das Optimum der verallgemeinerten Goldstein-Price-Funktion liegt bei $x = (0, -1, 0, -1, \ldots, 0, -1)$.

Um die Schwankungen von Messwerten, wie sie z.B. bei der Simulation auftreten, bei der Benutzung dieser Funktionen zu berücksichtigen, wird in einigen Experimenten ein normalverteilter **Fehler** mit Mittelwert 0 und Varianz $\sigma^2$ zum eigentlichen Funktionswert addiert. Die jeweils verwendeten Werte der Varianz sind dann bei den jeweiligen Experimenten angegeben.

## 2.3. Response Surface und räumliche Vorstellung

Die in dieser Arbeit vorgestellten Verfahren beruhen weitestgehend auf einer räumlichen Sicht der Probleme bzw. ist die räumliche Sicht für das Verständnis hilfreich.

Ackley-Funktion:

$$f_a(x) = -20 \cdot exp\left(-0,2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - exp\left(\frac{1}{n}\sum_{i=1}^{n} cos\left(2\pi x_i\right)\right) + 20 + e$$

Sinus-Funktion:

$$f_s(x) = \prod_{i=1}^{n} \sin x_i$$

Goldstein-Price-Funktion:

$$f_{gp}(x) = \Big(1 + (x_1 + x_2 + 1)^2 \cdot (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)\Big) \cdot \Big(30 + (2x_1 - 3x_2)^2 \cdot (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)\Big)$$

Schaffer-Funktion:

$$f_r(x) = \left(x_1^2 + x_2^2\right)^{0,25} \cdot \sin^2\left(50 \cdot \left(x_1^2 + x_2^2\right)^{0,1} + 1\right)$$

Tabelle 2.1.: Formeln der Ackley-, Sinus-, Goldstein-Price- und Schaffer-Funktion

Dies wird dadurch unterstützt, dass die vorgestellten Benchmarkfunktionen und -modelle im weiteren Verlauf dieser Arbeit mit ein oder zwei Parametern eingesetzt werden. Daher werden im weiteren Verlauf dieser Arbeit Begriffe verwendet, die eine räumliche Vorstellung erleichtern. Die bisher eingeführten Begriffe werden daher mit den in diesem Kontext gebräuchlichen Begriffen verknüpft.

Die in dieser Arbeit vorgestellten Verfahren suchen innerhalb eines Suchraums $\mathbf{W} \subset \mathbb{R}^n \times \mathbb{Z}^m$, der Menge der gültigen Modellkonfigurationen, nach Optima. Dazu wählen sie einzelne Modellkonfigurationen, genannt **Punkte**, aus. Die Auswertung eines Punkts $x$ - also die Berechnung von $f_M(x)$ mittels einer Auswertung des Modells - heißt **Messwert** oder **Response**. Aufgrund der stochastischen Eigenschaften einiger Modelle bzw. Funktionen können zu einem Punkt mehrere, unterschiedliche Messwerte bestimmt werden. Aus einer Verknüpfung der Punkte mit ihren Messwerte bzw. ihrem durchschnittlichem Messwert, entstehen so genannte **Response Surfaces**. Im Falle von ein- oder zweidimensionalen Funktionen können **Response Surfaces** als zwei- bzw. dreidimensionale Plots dargestellt werden.

## 2.4. Metamodell

Die Auswertung einer Funktion $f_M(x)$ zu einem Modell $M$ ist in vielen Fällen kosten- bzw. zeitintensiv. Daher ist es in bestimmten Fällen sinnvoll, nicht für alle Modellkonfigurationen den exakten Wert von $f_M(x)$ zu bestimmen, sondern auf eine Ap-

proximation zurückzugreifen, die weniger kosten- und zeitintensiv ist. Da aufgrund der Annahme von Black-Box-Modellen keine genaueren Informationen über die interne Struktur der Modelle vorliegen, können ausschließlich bereits durchgeführte Auswertungen des Modells die Grundlage für eine solche Approximation bilden. Ein Modell, das eine solche Approximation realisiert und mittels bekannter Punkte und deren Auswertungen angepasst wird, heißt Metamodell[1].

Zu einer Menge von Punkten $\mathbf{P} = \{p_1, \ldots, p_n\}$ gibt es eine Menge von Auswertungsvektoren $\mathbf{A} = \{a_1, \ldots, a_n\}$, so dass jedem Punkt $p_i$ der Auswertungsvektor $a_i$ zugeordnet ist, der die Ergebnisse aller Auswertungen des Punkts $p_i$ enthält.

**Definition 5 (Metamodell)** *Ein **Metamodell** $\hat{M}$ ist eine Funktion $f_{\hat{M}}(x)$, die eine Funktion $f_M(x)$ zu einem Modell $M$ auf der Basis einer Menge von Punkten $\mathbf{P}$ und deren Auswertungen $\mathbf{A}$ approximiert, ohne dabei das Modell $M$ zu benötigen.*

[1] Der Begriff des Metamodells ist in verschiedenen Anwendungsgebieten unterschiedlich genutzt. Im Bereich der Optimierung wird dieser Begriff in der Regel wie in dieser Arbeit im Sinne eines Ersatzmodells (siehe auch [Kle07a]) definiert.

# 3. Experimentdesigns

Die im letzten Kapitel vorgestellte Betrachtung von Modellen als Funktionen ermöglicht eine strukturierte Untersuchung der für die Zielfunktion wesentlichen Eigenschaften und Verhaltensweisen dieser Modelle. Eine verbreitete Möglichkeit zur Untersuchung solcher Funktionen ist der Einsatz von Experimentdesigns, wobei in der Statistik der Begriff Versuchspläne gebräuchlicher ist. Experimentdesigns sind Verfahren, die nach unterschiedlichen Kriterien eine Menge von Punkten bestimmen, die für ein festgelegtes Analyseziel in der Regel gut geeignet oder nach bestimmten Kriterien optimal ist. Dabei unterscheiden sich die Analyseziele vor allem dadurch, ob ein kleiner, lokaler Bereich um einen bereits bekannten Punkt oder das Verhalten der Funktion über den gesamten Wertebereich untersucht werden soll. Eine Auswahl an Experimentdesigns für beide Arten von Analysezielen wird im folgenden vorgestellt. Eine ausführlichere Behandlung von Experimentdesigns findet sich in der Literatur. Dabei beschränken sich einzelne Literaturstellen in der Regel auf bestimmte Anwendungsgebiete. Eine Übersicht über Experimentdesigns im Umgang mit Simulationsmodellen findet sich zum Beispiel in [Kle07b]. Der folgende Text beschränkt sich auf die in dieser Arbeit verwendeten Experimentdesigns.

**Definition 6 (Experimentdesign)** *Ein Experimentdesign (Design) ist eine Menge von n Punkten, die zur Untersuchung einer Funktion ausgesucht wurden. Die Punkte eines Experimentdesigns nennt man Designpunkte.*

Ein Design $X$ wird als $n \times k$-Matrix beschrieben, die zeilenweise die Koordinaten der Punkte enthält.

**Definition 7 (Gültigkeit)** *Ein Design $X$ ist gültig, wenn alle Punkte von $X$ im Wertebereich $\boldsymbol{W}$ von $f_M$ liegen.*

Zur sprachlichen Einheitlichkeit wird der Begriff des *Suchraums* genutzt. Der Suchraum $S$ ist die Schnittmenge eines $k$-dimensionalen Hyperkubus $H$ und des Wertebereichs $\mathbf{W}$. $H$ legt dabei den lokalen Bereich fest, der untersucht werden soll. Durch die Schnittmengenbildung wird dieser Bereich auf den gültigen Bereich begrenzt. Für globale Designs ist $S = \mathbf{W}$. Unabhängig von der Fragestellung und der Erstellung von lokalen bzw. globalen Designs erstrecken sich die Designs über den gesamten Suchraum $S$.

## 3.1. Designoptimalität

Es existiert eine Reihe von verschiedenen Optimalitätskriterien für Experimentdesigns. Zu diesen zählen zum Beispiel die A-, D-, E-, G- und die I-Optimalität. Eine ausführliche Beschreibung der Optimalitätskriterien findet sich zum Beispiel in [BD74] und [HS93]. In der Regel wird die D-Optimalität genutzt, da sie sowohl ein gutes Kriterium zur Bestimmung der Designqualität als auch einfach zu berechnen ist.

**Definition 8 (D-Optimalität)** *Für ein Design $X$ sei $D(X) = |X'X|$. Ein Design $X$ ist D-optimal, wenn $D$ für alle gültigen Designs nicht größer als $D(X)$ ist.*

Zusätzliche Kriterien sind die Orthogonalität und die Verteilung im Raum. Ein orthogonales Design ermöglicht, die Einflüsse aller Parameter zu bestimmen. Ein Design ist dann orthogonal, wenn die Werte jeder Dimension des Designs mit keiner anderen korrelieren, die Designvektoren also orthogonal zueinander sind. Für nicht orthogonale Designs gilt, dass kleinere Werte für die maximale Korrelation zweier Dimensionen als besser betrachtet werden. Die Verteilung eines Designs $D$ mit den Designpunkten $x_1, \ldots, x_n$ im Raum kann mittels der euklidischen Distanz wie folgt beurteilt werden:

$$v(D) = min_{(i=1,\ldots,n-1)\wedge(j=i+1,\ldots,n)}|x_i - x_j| \tag{3.1}$$

Dabei gilt, das größere Werte für $v$ einem besserem Design entsprechen, da die Punkte des Designs weiter auseinander liegen.

## 3.2. Erstellung von Designs

Eine erschöpfende Betrachtung aller Erstellungsmöglichkeiten kann angesichts der Menge verfügbarer Experimentdesigns nicht durchgeführt werden. Daher werden hier nur die Designs aufgeführt, die in späteren Kapiteln genutzt werden.

### 3.2.1. Faktorielle und teilfaktorielle Designs

Bei lokalen Untersuchungen werden in vielen Fällen faktorielle Designs (FD) oder teilfaktorielle Designs (TFD) genutzt. FDs sind leicht zu generierende Designs, die einem festgelegten Muster folgen und können daher nur eingesetzt werden, wenn der Suchraum ein Hyperkubus ist. In diesem Fall sind FDs auch D-optimal. Die hier aufgeführte Beschreibung orientiert sich im Wesentlichen an [MM02] und betrachtet nur zweistufige FDs. Mehrstufige FDs können analog erstellt werden. Weitere Informationen zu diesem Thema enthält [BHH05].

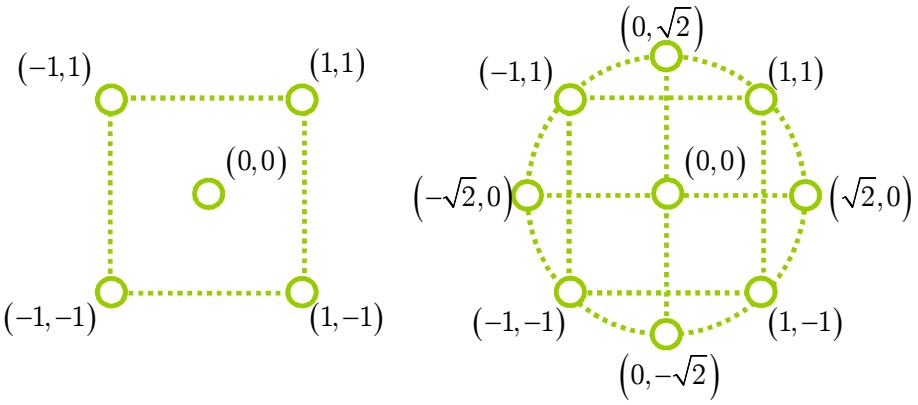



Abbildung 3.1.: Ein faktorielles und ein zentral zusammengesetztes Design um den Mittelpunkt $(0;0)$ mit Weite $w = 1$ und $k = 2$ Dimensionen

Ein faktorielles Design für $k$ Dimensionen besteht aus insgesamt $n = 2^k$ Punkten. Die Punkte eines faktoriellen Designs werden um einen Mittelpunkt $x$ mit einer Weite $w$ erzeugt, wobei für jede Dimension des Mittelpunkts die Werte $x_i - w$ und $x_i + w$ mit dem Wert $x_i$ des Mittelpunkts in dieser Dimension berechnet und alle möglichen Kombinationen erzeugt werden. Die Punkte dieses Designs bilden die Eckpunkte eines Quadrats, Würfels oder Hyperkubus (siehe Abbildung 3.1 links).

**Definition 9 (Faktorielles Design)** *Ein faktorielles Design mit Weite $w$ und $k$ Dimensionen um den Punkt $x = (x_1, \ldots, x_n)'$ entspricht der folgenden Punktmenge:*

$$\left\{ y \in \mathbb{R}^k | \forall i = 1, \ldots, k : y_i \in \{x_i - w, x_i + w\} \right\}$$

In der Regel werden die Wertebereiche für alle Dimensionen vor der Erstellung eines Designs normalisiert, so dass der Mittelpunkt im Nullpunkt liegt und die Werte jeder Dimension im Intervall [-1;1] liegen. Die Designs haben dann immer eine Weite $w = 1$. Diese Normalisierung wird im restlichen Text vorausgesetzt.

Da die Anzahl der Punkte in faktoriellen Designs mit der Anzahl der Dimensionen exponentiell ansteigt, wurden Methoden entwickelt, um die Anzahl der Punkte der Designs zu reduzieren. Dabei werden teilfaktorielle Designs (TFD) genutzt, die aus $n = 2^{k-p}$ Punkten bestehen, wobei zuerst ein faktorielles Design für die ersten $k - p$ Dimensionen erstellt wird. Die Werte der restlichen Dimensionen jedes Punkts ergeben sich dann anhand **erzeugender Muster** aus den bereits gewählten Werten. Bei normalisiertem Wertebereich berechnen sich die Werte der Dimensionen $k-p+1$ bis $k$ jeweils als Produkte mehrerer Werte der Dimensionen 1 bis $k - p$. Die ersten Dimensionen $k - p$ werden daher im Allgemeinen als Hauptdimensionen und die restlichen Dimensionen als Nebendimensionen bezeichnet. Für ein $2^{5-2}$-TFD könnte sich der Wert für die Dimensionen 4 und 5 als $x_4 = x_1 x_2$ und $x_5 = x_2 x_3$ oder

$x_4 = x_1x_3$ und $x_5 = x_2x_3$ berechnen lassen. Der Fall $x_4 = x_5 = x_2x_3$ ist ungünstig, da hier keine Unterschiede zwischen dem Einfluss von $x_4$ und $x_5$ abgeleitet werden können.

**Definition 10 (Erzeugendes Muster)** *Ein erzeugendes Muster $M$ für die Dimension $i \in \{k-p+1, \ldots, k\}$ für ein $k-p$ TFD mit normalisiertem Wertebereich ist das Produkt der Werte der bestimmenden Dimensionen.*

**Definition 11 (Länge eines erzeugenden Musters)** *Die Länge eines erzeugenden Musters ist die Anzahl der in diesem Muster vorkommenden Dimensionen.*

Ein erzeugendes Muster $M$ der Länge $l$, das durch Dimensionen $d_1, \ldots, d_l$ bestimmt wird, wird als $I_{d_1} \cdots I_{d_l}$ geschrieben. Im folgenden wird die verbreitete Schreibweise genutzt, bei der $-1$ durch $-$ und $+1$ durch $+$ dargestellt wird. Für ein $2^{3-1}$-TFD mit dem erzeugenden Muster $I_1I_2$ ergeben sich folgende Punkte:

| Punkt | Dimension | | | Identität |
|---|---|---|---|---|
| | 1 | 2 | 3 | I |
| 1 | - | - | + | + |
| 2 | - | + | - | + |
| 3 | + | - | - | + |
| 4 | + | + | + | + |

Eine weitere Schreibweise, mit der TFDs eindeutig beschrieben werden können, besteht darin alle Spaltenkombinationen anzugeben, deren Ergebnis der Identität $I$ entspricht. Diese Spaltenkombinationen heißen definierende Muster. Für das betrachtete $2^{3-1}$-TFD gilt $I = I_1I_2I_3$.

Die Qualität eines teilfaktoriellen Designs wird durch die **Auflösung** bestimmt, die zur besseren Unterscheidbarkeit in der Regel als römische Zahl angegeben wird.

**Definition 12 (Auflösung)** *Die Auflösung eines TFD ist die kleinste Länge aller möglichen definierenden Muster.*

Sie ist um eins größer als das kürzeste definierende Muster eines der Faktoren $k-p+1$ bis $k$, da die ersten $k-p$ Faktoren unabhängig sind und für das erzeugende Muster $M$ eines Faktors $I_l$ $I = MI_l$ gilt. Beispielsweise hat ein 5-2 TFD mit den erzeugenden Mustern $I_1I_2$ und $I_1I_3$ für die letzten beiden Dimensionen eine Auflösung von III. Für 4-2 TFD mit den erzeugenden Mustern $I_1I_2$ und $I_1I_3$ ergibt sich folgende Tabelle:

| Punkt | Dimension | | | | I |
|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | |
| 1 | - | - | + | - | + |
| 2 | - | + | - | + | + |
| 3 | + | - | - | - | + |
| 4 | + | + | + | + | + |

Aus dieser Tabelle wird deutlich, dass das kürzeste erzeugende Muster für die vierte Dimension $I_2$ ist. Die Auflösung dieses Designs ist daher II. TFDs mit einer kleineren Auflösung als III werden in der Regel nicht betrachtet, da hier nicht die Einflüsse aller Hauptdimensionen von denen der Nebendimensionen getrennt werden können. Die TFDs mit 3 bis 10 Dimensionen in Tabelle 3.1 basieren auf [MM02].

### 3.2.2. Zentral zusammengesetzte Designs

Das zentral zusammengesetzte Design (ZZD) (siehe Abbildung 3.1 rechts) entspricht einem faktoriellen Design, das um einige Punkte erweitert wurde. Für jede Dimension werden zwei zusätzliche Punkte erzeugt, die jeweils in dieser Dimension um $\pm\sqrt{2 \cdot w}$ vom Mittelpunkt verschoben werden.

**Definition 13 (Zentral zusammengesetztes Design)** *Ein zentral zusammengesetztes Design mit Weite $w$ und $k$ Dimensionen um den Punkt $x = (x_1, \ldots, x_n)'$ ist durch die folgende Punktmenge festgelegt:*

$$\left\{y \in \mathbb{R}^k | \left(\forall i = 1, \ldots, k : y_i \in \left\{x_i, x_i - \sqrt{2 \cdot w}, x_i + \sqrt{2 \cdot w}\right\}\right) \wedge \left(|\{i | y_i \neq x_i\}| = 1\right)\right\}$$
$$\cup$$
$$\left\{y \in \mathbb{R}^k | \forall i = 1, \ldots, k : y_i \in \{x_i - w, x_i + w\}\right\}$$

Die Gesamtzahl der Punkte eines ZZD ist also $n = 2^k + 2 \cdot k$. Bei zwei Faktoren liegen alle Punkte dieses Designs auf einem Kreis mit Radius $\sqrt{2 \cdot w}$. Bei mehr Faktoren liegen die Punkte auf entsprechenden Kugeln bzw. Hyperkugeln.

### 3.2.3. Computer Generierte Designs

FDs und ZZDs sind D-optimale Designs, die mit geringem Aufwand zu erstellen sind. Sie unterliegen jedoch Einschränkungen z.B. in Bezug auf den Suchraum. Liegt nach der Generierung eines Designs mindestens ein Punkt außerhalb des Suchraums kann dieses Design nicht genutzt werden. Computer Generierte Designs (CGD) können sich an solche Situationen anpassen und entsprechende Designs erstellen.

CGD sind Algorithmen, die in der Regel in drei Schritten ein möglichst gutes Design der Größe $n$ erstellen. Im ersten Schritt wird eine Menge von Kandidatenpunkten

| k | Design | Auflösung | Anzahl Punkte | Generierende Muster |
|---|---|---|---|---|
| 3 | $2^{3-1}$ | III | 4 | $I_3 = I_1 I_2$ |
| 4 | $2^{4-1}$ | IV | 8 | $I_4 = I_1 I_2 I_3$ |
| 5 | $2^{5-1}$ | V | 16 | $I_5 = I_1 I_2 I_3 I_4$ |
| | $2^{5-2}$ | III | 8 | $I_4 = I_1 I_2$, $I_5 = I_1 I_3$ |
| 6 | $2^{6-1}$ | VI | 32 | $I_6 = I_1 I_2 I_3 I_4 I_5$ |
| | $2^{6-2}$ | IV | 16 | $I_5 = I_1 I_2 I_3$, $I_6 = I_2 I_3 I_4$ |
| | $2^{6-3}$ | III | 8 | $I_4 = I_1 I_2$, $I_5 = I_1 I_3$, $I_6 = I_2 I_3$ |
| 7 | $2^{7-1}$ | VII | 64 | $I_7 = I_1 I_2 I_3 I_4 I_5 I_6$ |
| | $2^{7-2}$ | IV | 32 | $I_6 = I_1 I_2 I_3 I_4$, $I_7 = I_1 I_2 I_4 I_5$ |
| | $2^{7-3}$ | IV | 16 | $I_5 = I_1 I_2 I_3$, $I_6 = I_1 I_3 I_4$, $I_7 = I_2 I_3 I_4$ |
| | $2^{7-4}$ | III | 8 | $I_4 = I_1 I_2$, $I_5 = I_1 I_3$, $I_6 = I_2 I_3$, $I_7 = I_1 I_2 I_3$ |
| 8 | $2^{8-1}$ | VIII | 128 | $I_8 = I_1 I_2 I_3 I_4 I_5 I_6 I_7$ |
| | $2^{8-2}$ | V | 64 | $I_7 = I_1 I_2 I_3 I_4$, $I_8 = I_1 I_2 I_5 I_6$ |
| | $2^{8-3}$ | IV | 32 | $I_6 = I_1 I_2 I_3$, $I_7 = I_1 I_2 I_4$, $I_8 = I_2 I_3 I_4 I_5$ |
| | $2^{8-4}$ | IV | 16 | $I_5 = I_1 I_2 I_3$, $I_6 = I_1 I_2 I_4$, $I_7 = I_1 I_3 I_4$, $I_8 = I_2 I_3 I_4$ |
| 9 | $2^{9-2}$ | VI | 128 | $I_8 = I_1 I_3 I_4 I_6 I_7$, $I_9 = I_2 I_3 I_5 I_6 I_7$ |
| | $2^{9-3}$ | IV | 64 | $I_7 = I_1 I_2 I_3 I_4$, $I_8 = I_1 I_3 I_5 I_6$, $I_9 = I_3 I_4 I_5 I_6$ |
| | $2^{9-4}$ | IV | 32 | $I_6 = I_1 I_2 I_3 I_5$, $I_7 = I_1 I_2 I_4 I_5$, $I_8 = I_1 I_3 I_4 I_5$, $I_9 = I_2 I_3 I_4 I_5$ |
| | $2^{9-5}$ | III | 16 | $I_5 = I_1 I_2 I_3$, $I_6 = I_1 I_2 I_4$, $I_7 = I_1 I_3 I_4$, $I_8 = I_2 I_3 I_4$, $I_9 = I_1 I_2 I_3 I_4$ |
| 10 | $2^{10-3}$ | V | 128 | $I_8 = I_1 I_2 I_3 I_7$, $I_9 = I_1 I_3 I_4 I_5$, $I_{10} = I_1 I_3 I_4 I_6$ |
| | $2^{10-4}$ | IV | 64 | $I_7 = I_1 I_2 I_4 I_5$, $I_8 = I_1 I_3 I_4 I_6$, $I_9 = I_1 I_2 I_3 I_5$, $I_{10} = I_2 I_3 I_4 I_6$ |
| | $2^{10-5}$ | IV | 32 | $I_6 = I_1 I_2 I_3 I_4$, $I_7 = I_1 I_2 I_3 I_5$, $I_8 = I_1 I_2 I_4 I_5$, $I_9 = I_2 I_3 I_4 I_5$, $I_{10} = I_2 I_3 I_4 I_5$ |
| | $2^{10-6}$ | III | 16 | $I_5 = I_1 I_2 I_3$, $I_6 = I_1 I_2 I_4$, $I_7 = I_1 I_3 I_4$, $I_8 = I_2 I_3 I_4$, $I_9 = I_1 I_2 I_3 I_4$, $I_{10} = I_1 I_2$ |

Tabelle 3.1.: $2^{k-p}$ Teilfaktorielle Designs mit 3 bis 10 Dimensionen und maximal 128 Punkten [MM02]

erstellt. Dazu wird der Suchraum in ein Gitter aufgeteilt. Die Berührungspunkte des Gitters bilden die Menge der Kandidatenpunkte. Im zweiten Schritt werden $n$ Punkte ausgewählt, die das initiale Design bilden. Im dritten Schritt wird das initiale Design durch Austauschen von Punkten des Designs und der Kandidatenmenge iterativ bzgl. eines Optimalitätskriteriums (siehe Abschnitt 3.1) verbessert. Die bekanntesten Methoden sind die von Dykstra [Dyk71], Wynn-Mitchell-Methode [MM70, Wyn72] und die von Federov [Fed72]. Übersichten dieser und weiterer Methoden finden sich in [CN80] und [BD74].

### 3.2.4. Latin Hypercubes

Latin Hypercubes gehören zu den raumfüllenden Designs und werden z.B. im Zusammenhang mit Kriging Metamodellen (siehe Abschnitt 4.3) verwendet. Bei dieser Art von Designs wird versucht, den Suchraum möglichst gut bzw. gleichmäßig mit Designpunkten abzudecken [SWN03, Cio02]. Latin Hypercubes gehen dabei von diskreten Wertebereichen mit $m$ Werten aus. Zur Diskretisierung kontinuierlicher Wertebereiche wird der Wertebereich für jede Dimension in $m$ Teilbereiche eingeteilt. Die genauen Werte für jeden Designpunkt werden nach der Erstellung des diskreten Designs zufällig und in der Regel gleichverteilt aus dem entsprechenden Teilbereich gewählt. Für jeden Designpunkt eines Latin Hypercubes gilt, dass die Auswahl der Bereiche über eine Permutation des Vektors $(1, 2, \ldots, m)$ bestimmt wird. Die Größe eines Latin Hypercubes ist daher durch die Anzahl der Permutationen begrenzt. Latin Hypercubes wurden zuerst in [MBC79] beschrieben. Eine Übersicht über verschiedene Latin Hypercube Designs ist in [YLS00] zu finden.

#### Orthogonale Latin Hypercubes

Orthogonale Latin Hypercubes (OLH) [Ye98b, Ye98a] haben ein fest vorgegebenes Konstruktionsschema und garantieren eine gleichmäßige Verteilung der Designpunkte im Suchraum. OLH unterliegen daher einigen Einschränkungen. Für die Anzahl der Teilbereiche $m$ muss ein $l$ mit $m = 2^l + 1$ existieren und die Anzahl der Designpunkte $n$ muss kleiner oder gleich $m$ sein. OLH erzeugen $m$ Punkte von denen anschließend $n$ ausgewählt werden müssen, wenn das Design weniger als $m$ Designpunkte enthalten soll.

Für die Berechnung eines OLH wird die Permutationsmatrix $\mathbf{A_i}$ benötigt:

$$\mathbf{A_i} = \underbrace{\mathbf{I} \otimes \cdots \otimes \mathbf{I}}_{l-i} \otimes \underbrace{\mathbf{R} \otimes \cdots \otimes \mathbf{R}}_{i}$$

mit

$$\mathbf{I} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \mathbf{R} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \tag{3.2}$$

$\otimes$ ist das Kronecker-Produkt zweier Matritzen. Darüber hinaus werden die Vektoren $a_i$ benötigt:

$$a_i = b_1 \otimes b_2 \otimes \cdots \otimes b_l$$

mit

$$b_j = \begin{cases} j = i & : \begin{pmatrix} -1 \\ 1 \end{pmatrix} \\ \text{sonst} & : \begin{pmatrix} 1 \\ 1 \end{pmatrix} \end{cases} \tag{3.3}$$

Aus den Formeln 3.2 und 3.3 lassen sich die Menge $\mathbf{G}$ aller $\mathbf{A_i}$ und die Menge $\mathbf{H}$ aller $a_i$ mit $i = 1, \ldots, l$ erstellen. Sei nun $\mathbf{\Lambda_1} = \{\mathbf{G_1}, \mathbf{G_2}, \ldots, \mathbf{G_k}\}$ eine Teilmenge von $\mathbf{G}$ und $\mathbf{\Lambda_2} = \{\mathbf{H_1}, \mathbf{H_2}, \ldots, \mathbf{H_k}\}$ eine Teilmenge von $\mathbf{H}$. Darüber hinaus muss für alle Paare $\mathbf{G_x}, \mathbf{G_y} \in \mathbf{\Lambda_1}$ und $\mathbf{H_u}, \mathbf{H_v} \in \mathbf{\Lambda_2}$ folgende Gleichung erfüllt sein:

$$\mathbf{G_1 G_2}(\mathbf{H_1} \times \mathbf{H_2}) = -(\mathbf{H_1} \times \mathbf{H_2}) \tag{3.4}$$

wobei $\times$ für das Schur-Produkt zweier Matritzen steht.

$e$ sei der Vektor $(1, 2, \ldots, 2^l)$. $\mathbf{M}$ sei eine $2^l \times l$ Matrix, deren $j$-te Spalte $\mathbf{G_j}$ entspricht, und $\mathbf{S}$ sei eine $2^l \times l$ Matrix, deren $j$-te Spalte $\mathbf{H_j}$ entspricht. Die Matrix $\mathbf{T}$ ist das Hadamard Produkt von $\mathbf{M}$ und $\mathbf{S}$. Die Matrix $\mathbf{O}$, die zeilenweise die Permutationen für die $n = 2^l + 1$ Designpunkte enthält, schreibt sich dann als:

$$\mathbf{O} = \begin{pmatrix} \mathbf{T} \\ \mathbf{0'_n} \\ -\mathbf{T} \end{pmatrix}$$

mit

$$\mathbf{0_n} = (0, \ldots, 0)' \tag{3.5}$$

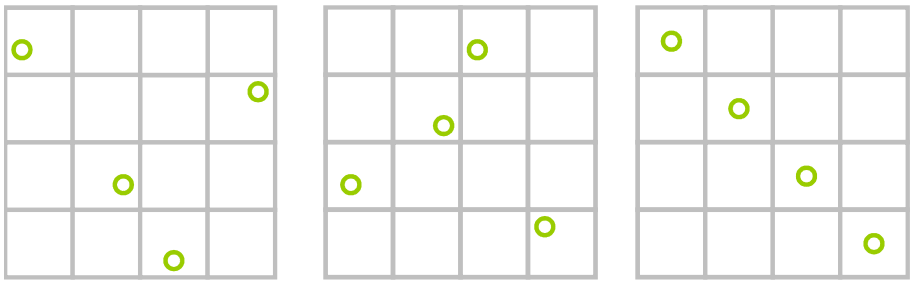


Abbildung 3.2.: Beispiele für Latin Hypercube Samplings mit $k = 2$ Dimensionen und $n = 4$ Punkten

Es existieren weitere Möglichkeiten zur Erstellung von OLH, die z.B. in der Erstellung von $\mathbf{\Lambda_1}$ und $\mathbf{\Lambda_2}$ abweichen. Die Verfahren sind jedoch dadurch beschränkt, dass $m$ durch $2^l + 1$ vorgegeben ist, und $m$ Designpunkte berechnet werden. Eine Bestimmung von weniger Designpunkte wird in der Regel durch die Auswahl der ersten $n$ Punkte erreicht.

### Latin Hypercube Sampling

Beim Latin Hypercube Sampling (LHS) [MBC79] werden $n$ Designpunkte über $m = n$ Teilbereichen für jeden Parameter erstellt. Hierzu wird aus allen möglichen Permutationen des Vektors $e = (1, 2, \ldots, m)$ zufällig $n$ ausgewählt. Die allgemeine Beschreibung dieser Methode ist daher deutlich einfacher als die der OLH. Im Gegensatz zu den OLH wird jedoch keine bestimmte Eigenschaft garantiert, so dass schlechte Experimentdesigns möglich sind, bei denen z.B. alle Designpunkte in der Nähe der Diagonalen liegen (siehe Abbildung 3.2 rechts). Eine Bewertung und das Verwerfen entstandener Designs kann daher sinnvoll sein.

# 4. Approximation von Funktionen

Die Auswertung von ereignisdiskreten Modellen ist in der Regel kostenintensiv. Häufig ist daher eine intensive Untersuchung der durch ein solches Modell gegebenen Funktion unter Berücksichtigung eines festgelegten Zeitrahmens nicht möglich. In diesem Fall ist es sinnvoll, nicht alle Untersuchungen direkt an $f$ durchzuführen, sondern eine Ersatzfunktion $g$ zu ermitteln, die $f$ unter Berücksichtigung des Untersuchungsziels möglichst gut approximiert. Mit dieser Funktion werden dann weitere intensive Untersuchungen durchgeführt und die Ergebnisse an der Originalfunktion $f$ validiert.

Grundsätzlich können beliebige Terme zur Erstellung einer mathematischen Ersatzfunktion genutzt werden. In der Praxis hat sich eine Reihe von parametrierbaren Ersatzfunktionen etabliert, zu denen zum Beispiel Lineare Regressionsmodelle, Kriging Metamodelle und Splines gehören. Diese Modelle ermöglichen, anhand weniger bekannter Punkte der Funktion $f$ ein Metamodell zu erstellen. Lineare Regressionsmodelle und Kriging Metamodelle werden in den folgenden Abschnitten genauer beschrieben. Weitere Informationen zu Regressions- und Metamodellen finden sich zum Beispiel in [MM02] und [KS00].

Lineare Regressionsmodelle (LR) werden vor allem zur Approximation lokaler Bereiche von Funktionen eingesetzt. LR höherer Ordnung werden seltener eingesetzt, da die Anzahl der benötigten Punkte und damit die Anzahl der Auswertungen mit der Ordnung exponentiell ansteigt. Darüber hinaus setzen lineare Regressionsmodelle voraus, dass für jeden Punkt $x$ genau ein Funktionswert $y_x = f(x)$ existiert. Da dies bei stochastischen Funktionen nicht gegeben ist und der Erwartungswert von $f(x)$ in der Regel nicht bekannt ist, wird hier als Schätzer für $E[f(x)]$ der Mittelwert der gemessenen Funktionswerte verwendet. Die Anpassung eines lineares Regressionsmodells erster bzw. zweiter Ordnung wird in den beiden folgenden Unterkapiteln beschrieben und orientiert sich an der Beschreibung in [MM02].

## 4.1. Lineare Regressionsmodelle erster Ordnung

LR erster Ordnung approximieren auf der Basis der Designpunkte eine Hyperebene. Diese Hyperebene dient z.B. zur Bestimmung der Richtung des steilsten An- oder Abstiegs an der Stelle des Mittelpunkts $x_c$ des Designs und stellt dementsprechend

eine Approximation der Steigung im Punkt $x_c$ dar. Die Hyperebene wird dabei wie folgt berechnet:

$$\hat{y} = \beta_0 + \sum_{i=1}^{k} (\beta_i \cdot x_{c,i}) \tag{4.1}$$

Zur Bestimmung des linearen Regressionsmodells erster Ordnung (siehe Formel 4.1) müssen zuerst die Designpunkte festgelegt und ausgewertet werden. In der Regel werden hierzu faktorielle oder teilfaktorielle Designs (siehe Kapitel 3.2.1) mit der vorgegebenen Breite des lokalen Suchbereichs $w$ und dessen Mittelpunkt $x$ erstellt. Dabei werden $n \geq k + 1$ Designpunkte mit dem zugehörigen Messwert benötigt, um die Regressionskoeffizienten $\beta_0$ bis $\beta_k$ zu bestimmen. Mit den Designpunkten $x_i = (x_{i1}, \ldots, x_{ik})'$ und den zugehörigen Messwerten $y_i$ mit $i = 1, \ldots, n$ lassen sich die in Formel 4.2 beschriebenen Matrizen bilden, wobei der Vektor $\boldsymbol{\beta}$ und der Vektor $\varepsilon$ unbekannt sind. Mit der Methode der kleinsten Fehlerquadrate kann Formel 4.2 berechnet und damit die Regressionskoeffizienten $\boldsymbol{\beta}$ mit minimalen $\varepsilon'\varepsilon$ bestimmt werden:

$$\mathbf{y} = \mathbf{X} \cdot \boldsymbol{\beta} + \varepsilon$$

mit

$$\mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}, \mathbf{X} = \begin{pmatrix} 1 & x_{11} & \ldots & x_{1k} \\ 1 & x_{21} & \ldots & x_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & \ldots & x_{nk} \end{pmatrix}, \boldsymbol{\beta} = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_k \end{pmatrix}, \varepsilon = \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{pmatrix} \tag{4.2}$$

Die Richtung des steilsten Anstiegs kann aus dem Vektor $\beta$ bestimmt werden und entspricht dem Vektor $(\beta_1, \ldots, \beta_k)'$.

## 4.2. Lineare Regressionsmodelle zweiter Ordnung

Ein lineares Regressionsmodell zweiter Ordnung besteht aus einem Polynom zweiten Grades mit $1 + (k^2 + 3 \cdot k)/2$ Regressionskoeffizienten, die gleichzeitig die Mindestanzahl auszuwertender Punkte festlegen. Es basiert auf folgendem Polynom zweiten Grades:

$$\mathbf{y} = \beta_0 + \sum_{i=1}^{k} (\beta_i \cdot x_i) + \sum_{i=1}^{k} \left(\beta_{i,i} \cdot x_i^2\right) + \sum_{i=1}^{k} \sum_{j=i+1}^{k} (\beta_{i,j} \cdot x_i \cdot x_j) + \varepsilon \tag{4.3}$$

Die Regressionskoeffizienten $\beta_0$ bis $\beta_k$ entsprechen denen des Regressionsmodells erster Ordnung. Die weiteren Regressionskoeffizienten $\beta_{i,i}$ mit $i = 1, \ldots, k$ sind die Koeffizienten der Quadrate aller Faktoren und die Regressionskoeffizienten $\beta_{i,j}$ mit $i = 1, \ldots, k-1$ und $j = i+1, \ldots, k$ sind die Koeffizienten für alle paarweisen Produkte der Faktoren. Da es sich um ein lineares Regressionsmodell handelt, werden die paarweisen Kombinationen der Variablen als eigenständige Variablen interpretiert und das Modell so linearisiert. Die dafür notwendige Ersetzung wird wie folgt durchgeführt:

- $\beta_{k+1} := \beta_{1,1}, \ldots, \beta_{2k} := \beta_{k,k}, \beta_{2k+1} := \beta_{1,2}, \ldots$
- $x_{k+1} := x_1 \cdot x_1, \ldots, x_{2k} = x_k \cdot x_k, x_{2k+1} := x_1 \cdot x_2, \ldots$

Dadurch entsteht ein lineares Regressionsmodell erster Ordnung mit $(k^2 + 3 \cdot k)/2$ Variablen und $1 + (k^2 + 3 \cdot k)/2$ Regressionskoeffizienten. Zur Bestimmung der Regressionskoeffizienten kann die Methode der kleinsten Fehlerquadrate durchgeführt werden. Mit den ermittelten Regressionskoeffizienten kann nach Rückersetzung die in Formel 4.4 dargestellte polynomielle Funktion zweiten Grades erstellt werden, die als Approximation des Response Surface dient:

$$\hat{\mathbf{y}} = \beta_0 + x'\boldsymbol{\beta} + x'\hat{\mathbf{B}}x$$

mit

$$\boldsymbol{\beta} = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_k \end{pmatrix}, \hat{\mathbf{B}} = \begin{pmatrix} \beta_{11} & \beta_{12}/2 & \ldots & \beta_{1k}/2 \\ & \beta_{22} & \ldots & \beta_{2k}/2 \\ & & \ddots & \vdots \\ sym. & & & \beta_{kk} \end{pmatrix}, \mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_k \end{pmatrix} \tag{4.4}$$

Aus den Matrizen $\hat{\mathbf{B}}$ und $\boldsymbol{\beta}$ lässt sich der stationäre Punkt $x_s$ wie folgt bestimmen:

$$x_s = -\frac{1}{2}\hat{\mathbf{B}}^{-1}\boldsymbol{\beta} \tag{4.5}$$

Die Art des stationären Punkts $x_s$ kann anhand der Eigenwerte der Matrix $\hat{\mathbf{B}}$ bestimmt werden. Sind alle Eigenwerte negativ, so handelt es sich um ein Maximum, sind alle Eigenwerte positiv, handelt es sich um ein Minimum und in allen anderen Fällen ist $x_s$ ein Sattelpunkt. Da die Linearen Regressionsmodelle zweiter Ordnung mehr Punkte als die Linearen Regressionsmodelle erster Ordnung benötigen, werden hier in der Regel zentral zusammengesetzte Designs (siehe Kapitel 3.2.2) für die Bestimmung der Punkte genutzt.

## 4.3. Kriging Metamodelle

Kriging Metamodelle (KM) sind im Vergleich zu den bisher beschriebenen linearen Regressionsmodellen deutlich flexibler und weiteren Modellen wie Gridding und Triangulation [RBL93] in vielen Fällen überlegen. Sie wurden ursprünglich von Krige in seiner Masterarbeit [Kri51] beschrieben. Matheron untersuchte die KM systematisch und erstellte erste detaillierte Übersichten [Mat71], bei denen zuerst der Begriff Krigeage genutzt wurde. Der Begriff Kriging bzw. Kriging Metamodell setzte sich erst im Laufe der Zeit durch. Der Bekanntheitsgrad von KM ist vor allem im Bereich der Geostatistik seitdem stark gestiegen und wegen der weiter steigenden Rechenleistung von Computern werden KM auch für andere Bereiche immer interessanter. Dies lässt sich leicht mit der in den letzten Jahren stark gestiegenen Anzahl an Publikationen zu diesem Thema belegen. 1986 hat sich Wu mit einer mathematischen Einordnung der Kriging-Methode beschäftigt und präsentierte Ansätze, die diese Methode mathematisch besser handhabbar machen [Wu86]. Eine praxisnahe Betrachtung wurde 1989 in [SWMW89] durchgeführt. Im Jahr 1993 wurden die Grundlagen der KM von Cressie in [Cre93] erneut zusammenfasst. 2004 befasste sich Kleijnen mit KM und gab dort eine Übersicht über den aktuellen Stand von KM. Darüber hinaus schneidet er kurz Themen an, die seiner Meinung nach bei der Nutzung von KM für ereignisdiskrete Simulationsmodelle genauer untersucht werden sollten:

- Welche Designs sollten genutzt werden?
- Wie viel Zeit sollte in Replikationen bei Simulationsmodellen investiert werden?
- Wie können mehrere Ausgaben eines Modells umgesetzt werden?
- Wie verhalten sich KM im Vergleich zu anderen Metamodellen?

2009 wurde von Forrester und Keane eine Übersicht Metmodell basierter Verfahren vorgestellt [FK09]. Diese stellt insbesondere die Ähnlichkeit von KM zu anderen Verfahren heraus und betrachtet die sich daraus ergebenden Vor- und Nachteile.

Wegen der vielen Publikationen zum Thema KM könnte der Eindruck entstehen, dass KM die Lösung für praktisch alle Probleme im Bereich der Metamodelle ist. Um diesem Eindruck entgegen zu wirken, wird auf [WBJS03] verwiesen. Dort wurden KM mit neuronalen Netzen und Evolutionäre Algorithmen (EA) verbunden. Empirische Untersuchungen ergaben jedoch keine Verbesserung gegenüber dem reinen Einsatz von EA.

KM wurden ursprünglich für deterministische Funktionen entworfen. Dort haben sie gegenüber vielen anderen Verfahren den Vorteil, dass sie den Funktionswert an bekannten Punkten exakt wiedergeben, weshalb KM auch als **Best Mean Squared Predictor** bezeichnet wird. Bei KM wird davon ausgegangen, dass sich die Funktion

$f(x)$ durch eine Funktion $y(x)$ approximieren lässt, die sich aus einem ggf. bekannten Teil $\mu(x)$ und einer Abweichung von diesem Teil $\varepsilon(x)$ zusammensetzt:

$$y(x) = \mu(x) + \varepsilon(x) \tag{4.6}$$

Dabei wird bei KM davon ausgegangen, dass die Abweichung $\varepsilon(x)$ eine normalverteilte Zufallsvariable mit Mittelwert 0 ist.

Anhand der Funktion $\mu(x)$ lassen sich die drei Arten einfaches, normales und universelles Kriging unterscheiden. Beim einfachen Kriging wird davon ausgegangen, dass $\mu(x)$ eine Konstante und vorher bekannt ist. Eine Berechnung dieser Konstanten, die dann in der Regel mit $\mu$ gekennzeichnet wird, ist dementsprechend nicht notwendig, so dass nur der zweite Teil von 4.6 approximiert werden muss. Beim normalen KM wird $\mu(x)$ ebenfalls als konstant betrachtet, der Wert ist jedoch nicht vorher bekannt und wird aus den bereits ausgewerteten Punkten berechnet. Beim universellen Kriging wird $\mu(x)$ hingegen als eine Summe von $l$ beliebig wählbaren Funktionen $g_i(x)$ mit $i = 1, \ldots, l$ aufgefasst. So ist das universelle Kriging insgesamt mächtiger, da es die oben genannte Voraussetzung für $\varepsilon(x)$ herstellen kann. Es benötigt jedoch zusätzliche Informationen über den Verlauf von $f$ bzw. muss sowohl die Art als auch den Verlauf der Funktionen für $\mu(x)$ bestimmen. Diese Arbeit beschränkt sich auf die am weitesten verbreitete Variante das normale Kriging.

Basierend auf diesen drei Arten existieren verschiedene Varianten, die auf spezielle Anwendungsgebiete zugeschnitten sind. Zu diesen zählen z.B. das Median Polish Kriging [Cre93, Ber01] und das Block Kriging [MT02]. Für das Median Polish Kriging müssen alle Eckpunkte eines Gitters ausgewertet werden. Damit ist dieses Verfahren nur in Situationen geeignet, bei denen zuerst die Messpunkte bestimmt und exakt eingehalten werden können. In Bereichen wie der Optimierung kann dieses Verfahren daher nicht eingesetzt werden. Das Block Kriging wurde im Kontext landwirtschaftlicher Messungen entwickelt und liefert statt einer Schätzung an einzelnen Punkten Schätzwerte für Bereiche. Es wird im ursprünglichen Kontext vor allem zur Schätzung des durchschnittlichen Gehalts bestimmter Substanzen auf Feldern eingesetzt. Darüber hinaus existieren Varianten um mit ständig steigenden Datenmengen umgehen zu können. Eines davon ist das adaptive Kriging [WS07], bei dem die Daten an der gleichen Stelle, aber an aufeinander folgenden Zeitpunkten gemessen werden. Zur Reduktion der Datenmenge wird hier ein rekursiver Schätzer entwickelt. KM werden mittlerweile in vielen verschiedenen Anwendungsbereichen wie z.B. der Bildbearbeitung [SBB06] und der Schätzung von Ableitungen und Integralen [VW05] eingesetzt.

### 4.3.1. Approximation der Funktion $f$

Der Wert von $y(x)$ kann nicht exakt bestimmt werden, da die zu approximierende Funktion $f$ nicht oder nicht vollständig bekannt ist. Sie wird daher durch $\hat{y}(x)$

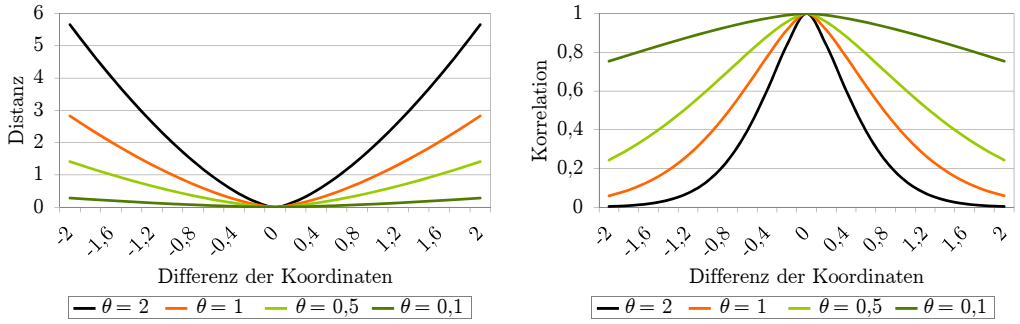



Abbildung 4.1.: Distanz und Korrelation in einem eindimensionalen Kriging Metamodell in Abhängigkeit von $\theta$ bei konstantem $\rho = 1,5$

auf Basis der bekannten Punkte angenähert. Dabei haben Punkte, die näher an $x$ liegen, einen größeren Einfluss auf den Wert von $\hat{y}(x)$ als Punkte, die eine größere Entfernung zu $x$ aufweisen. Da die unterschiedlichen Dimensionen in der Regel einen unterschiedlich starken Einfluss auf den Funktionswert ausüben, wird dies nicht auf der Basis der euklidischen Distanz sondern der gewichteten Distanz $d(x_1, x_2)$ (Kriging Distanz) zweier Punkte $x_1$ und $x_2$ durchgeführt:

$$d(x_1, x_2) = \sum_{h=1}^{k} \theta_h \left| x_{1,h} - x_{2,h} \right|^{\rho_h} \quad \text{mit} \quad \theta_h \geq 0; \rho_h \in [1, 2] \tag{4.7}$$

Hierbei gibt der Vektor $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_k)'$ die Gewichtung der einzelnen Dimensionen und der Vektor $\boldsymbol{\rho} = (\rho_1, \ldots, \rho_k)'$ die Weichheit des Funktionsverlaufs für die einzelnen Dimensionen an. Die Bestimmung der Vektoren $\boldsymbol{\theta}$ und $\boldsymbol{\rho}$ wird in Abschnitt 4.3.2 beschrieben. Die Korrelation zweier Punkte $x_1$ und $x_2$ berechnet sich dann wie folgt:

$$Corr\left[x_1, x_2\right] = e^{-d(x_1, x_2)} \tag{4.8}$$

Die Korrelation zweier Punkte ist größer als 0 und kleiner gleich 1. Für den Fall $\theta_h > 0$ für $h \in \{1, \ldots, k\}$ gilt, dass die Aussage $x_1 = x_2$ äquivalent zu $Corr\left[x_1, x_2\right] = 1$ ist. Der Einfluss von $\boldsymbol{\theta}$ und $\boldsymbol{\rho}$ auf die Distanz und die Korrelation in einem Kriging Metamodell ist in den Abbildungen 4.1 und 4.2 dargestellt.

Die Korrelation aller $n$ bekannten Punkte zueinander wird in der $n \times n$-Matrix $\boldsymbol{R}$ festgehalten:

$$\boldsymbol{R} = \begin{pmatrix} Corr\left[x_1, x_1\right] & = 1 & \ldots & Corr\left[x_1, x_n\right] \\ \vdots & & \ddots & \vdots \\ Corr\left[x_n, x_1\right] & & \ldots & Corr\left[x_n, x_n\right] & = 1 \end{pmatrix} \tag{4.9}$$

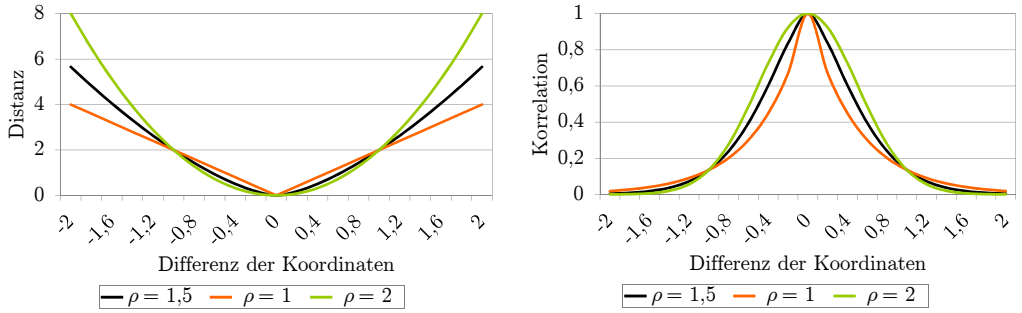



Abbildung 4.2.: Distanz und Korrelation in einem eindimensionalen Kriging Metamodell in Abhängigkeit von $\rho$ bei konstantem $\theta = 2$

Die Korrelation aller bekannten Punkte zu einem Punkt $\boldsymbol{x}$ wird im Korrelationsvektor $\boldsymbol{r}(\boldsymbol{x}) = (Corr\left[\boldsymbol{x}, \boldsymbol{x}_1\right], \ldots, Corr\left[\boldsymbol{x}, \boldsymbol{x}_n\right])'$ festgehalten.

Entsprechend Formel 4.6 wird angenommen, dass der Funktionswert eines Punkts $\boldsymbol{x}$ mit $y(\boldsymbol{x}) = \mu(\boldsymbol{x}) + \varepsilon(\boldsymbol{x})$ berechnet werden kann. Mit $\hat{\mu} = \frac{\mathbf{1}_n' \boldsymbol{R}^{-1} \boldsymbol{y}}{\mathbf{1}_n' \boldsymbol{R}^{-1} \mathbf{1}_n}$ und dem n-Vektor $\mathbf{1}_n = (1, \ldots, 1)'$ erhält man eine Approximation des Erwartungswerts $\mu$ der Funktion $y(x)$. Die Funktionswerte aller Punkte des KM werden im Vektor $\boldsymbol{y} = (y(\boldsymbol{x}_1), \ldots, y(\boldsymbol{x}_n))'$ festgehalten. Mit $\boldsymbol{y} - \mathbf{1}_n\hat{\mu}$ erhält man einen Vektor der Abweichungen vom erwarteten Mittelwert $\hat{\mu}$. Die Abweichung des Punkts $\boldsymbol{x}$ vom Mittelwert wird über die Korrelation zu den bekannten Punkten als $\boldsymbol{r}(\boldsymbol{x})'\boldsymbol{R}^{-1}(\boldsymbol{y} - \mathbf{1}_n\hat{\mu})$ bestimmt. Der Funktionswert $y(\boldsymbol{x})$ kann nun wie folgt durch $\hat{y}(\boldsymbol{x})$ approximiert werden:

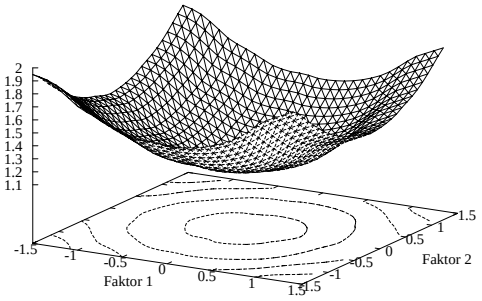
$$\hat{y}(\boldsymbol{x}) = \hat{\mu} + \boldsymbol{r}(\boldsymbol{x})'\boldsymbol{R}^{-1}(\boldsymbol{y} - \mathbf{1}_n\hat{\mu}) = \hat{\mu} + \boldsymbol{c}'\boldsymbol{r}(\boldsymbol{x}) \quad \text{mit} \quad \boldsymbol{c} = \boldsymbol{R}^{-1}(\boldsymbol{y} - \mathbf{1}_n\hat{\mu}) \tag{4.10}$$

Detaillierte Informationen zur Herleitung dieser Formel werden in [Cre93] aufgeführt. Abbildung 4.3 enthält zwei grafische Darstellungen der Schätzwerte für ein Kriging Metamodell in Abhängigkeit zu den Parametern $\boldsymbol{\rho}$ und $\boldsymbol{\theta}$. Die beiden Grafiken zeigen deutlich einen weicheren bzw. kantigeren Verlauf, der durch die unterschiedlichen Werte für $\boldsymbol{\rho}$ hervorgerufen wird, sowie eine stärkere bzw. schwächere Ausprägung der Schätzwerte, die durch die unterschiedlichen Werte für $\boldsymbol{\theta}$ hervorgerufen wird.

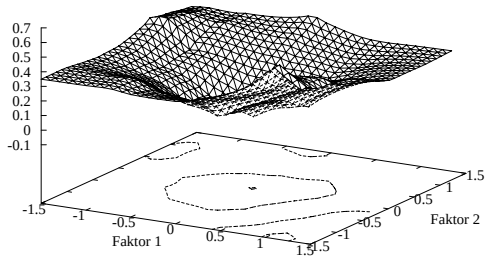
Dieser Schätzer ist ein bester Schätzer bzgl. der Abweichung an den bekannten Punkten, da diese konstruktionsbedingt gleich 0 ist (siehe [JSW98]). Zusätzlich lässt sich der erwartete quadrierte Fehler für jeden Punkt $\boldsymbol{x}$ wie folgt schätzen:

$$s^2(\boldsymbol{x}) = \sigma^2 \left(1 - \boldsymbol{r}(\boldsymbol{x})'\boldsymbol{R}^{-1}\boldsymbol{r}(\boldsymbol{x}) + \frac{\left(1 - \mathbf{1}_n'\boldsymbol{R}^{-1}\mathbf{1}_n\right)^2}{\mathbf{1}_n'\boldsymbol{R}^{-1}\mathbf{1}_n}\right) \tag{4.11}$$

Da $\sigma^2$ unbekannt ist wird dieser wie folgt abgeschätzt:




$\boldsymbol{\rho} = \mathbf{1}_2$ und $\boldsymbol{\theta} = \mathbf{2}_2$




$\boldsymbol{\rho} = \mathbf{1,5}_2$ und $\boldsymbol{\theta} = \mathbf{0.1}_2$

Abbildung 4.3.: Darstellung eines Kriging Metamodells für die Sinus-Funktion mit unterschiedlichen Werten für $\boldsymbol{\theta}$ und $\boldsymbol{\rho}$ basierend auf 16 gemessenen Punkten

$$\hat{\sigma}^2 = \frac{(\boldsymbol{y} - \mathbf{1}_n\hat{\mu})'\boldsymbol{R}^{-1}(\boldsymbol{y} - \mathbf{1}_n\hat{\mu})}{n} \tag{4.12}$$

Kriging Metamodelle bieten einen Schätzer $\hat{y}$ für den Funktionswert $f$ und einen Schätzer für den maximalen erwarteten Fehler $s$. Damit sind sie flexibler einsetzbar als zum Beispiel Lineare Regressionsmodelle, benötigen jedoch in der Regel mehr bekannte Punkte.

### 4.3.2. Bestimmung von $\theta$ und $\rho$ mittels Maximum-Likelihood

Zur Bestimmung der Vektoren $\boldsymbol{\theta}$ und $\boldsymbol{\rho}$ wird in der Regel das Maximum-Likelihood-Verfahren angewendet. Bei diesem Verfahren werden $\boldsymbol{\theta}$ und $\boldsymbol{\rho}$ über das Maximum der Likelihood-Funktion bestimmt. Die Likelihood-Funktion $l(\boldsymbol{\theta}, \boldsymbol{\rho}, \mu, \sigma^2)$ ist dabei wie folgt definiert:

$$l(\boldsymbol{\theta}, \boldsymbol{\rho}, \mu, \sigma^2) = \frac{e^{-\frac{(\boldsymbol{y}-\mathbf{1}_n\mu)'\boldsymbol{R}^{-1}(\boldsymbol{y}-\mathbf{1}_n\mu)}{2\sigma^2}}}{(2\pi\sigma^2)^{n/2}\sqrt{|\boldsymbol{R}|}} \tag{4.13}$$

Da $\mu$ und $\sigma^2$ unbekannt sind, werden sie durch Schätzwerte $\hat{\mu}$ und $\hat{\sigma}^2$ ersetzt. Des Weiteren ist zu beachten, dass die Matrix $\boldsymbol{R}$ und $\hat{\mu}$ abhängig von $\boldsymbol{\theta}$ und $\boldsymbol{\rho}$ sind. Für eine vollständig korrekte Schreibweise müssten $\boldsymbol{R}$ und $\hat{\mu}$ daher durch eine Funktion $R(\boldsymbol{\theta}, \boldsymbol{\rho})$ bzw $\hat{\mu}(\boldsymbol{\theta}, \boldsymbol{\rho})$ ersetzt werden, die die Matrix $\boldsymbol{R}$ und $\hat{\mu}$ für $\boldsymbol{\theta}$ und $\boldsymbol{\rho}$ berechnen. Zur besseren Lesbarkeit wird an dieser Stelle darauf verzichtet. Dadurch ergibt sich

die konzentrierte Maximum-Likelihood-Funktion:

$$l_{konzentriert}(\boldsymbol{\theta}, \boldsymbol{\rho}) = \left( \left( \frac{2\pi e}{n} \right)^{\frac{n}{2}} \left( \boldsymbol{X}' \boldsymbol{R}^{-1} \boldsymbol{X} \right)^{\frac{n}{2}} \sqrt{|\boldsymbol{R}|} \right)^{-1} \text{ mit } \boldsymbol{X} = \boldsymbol{y} - \mathbf{1}_n \frac{\mathbf{1}_n' \boldsymbol{R}^{-1} \boldsymbol{y}}{\mathbf{1}' \boldsymbol{R}^{-1} \mathbf{1}} \tag{4.14}$$

Da das Maximum der Funktion $l$ gesucht wird, können einige konstante Terme entfernt werden:

$$l_{vereinfacht}(\boldsymbol{\theta}, \boldsymbol{\rho}) = \sqrt{(\boldsymbol{X}' \boldsymbol{R}^{-1} \boldsymbol{X})^n \cdot |\boldsymbol{R}|}^{-1} \tag{4.15}$$

Aufgrund der Struktur der Likelihood-Funktion ist es einfacher den Logarithmus der Funktion zu berechnen:

$$\begin{aligned} l_{ln}(\boldsymbol{\theta}, \boldsymbol{\rho}) &= \ln(l_{vereinfacht}(\boldsymbol{\theta}, \boldsymbol{\rho})) \\ &= -\tfrac{1}{2} \left( n \cdot \ln \left( (\boldsymbol{y} - \mathbf{1}_n \mu)' \boldsymbol{R}^{-1} (\boldsymbol{y} - \mathbf{1}_n \mu) \right) + ln \, |\boldsymbol{R}| \right) \end{aligned} \tag{4.16}$$

Das Maximum der Funktion $l_{ln}$ entspricht dem Maximum der Funktion $l$ und muss mittels geeigneter Methoden bestimmt werden. Je nach Ansatz können hierzu spezielle Verfahren, wie sie z.B. in [JSW98] beschrieben werden, oder allgemeine Optimierungsverfahren, wie sie im folgenden Kapitel beschrieben werden, genutzt werden.

Neben der bisher beschriebenen, normalen Likelihood-Funktion existieren verschiedene weitere, die insbesondere Grenzfälle besser berücksichtigen sollen. So kann es bei der normalen Likelihood-Funktion der Fall sein, dass das Maximum an den Grenzen des Definitionsbereichs liegt, die Funktion aber nur sehr schwach in diesen Bereich ansteigt. In diesen Situationen werden ggf. einige Faktoren zu wenig bzw. zu viel berücksichtigt. Die nun vorgestellten Likelihood-Funktionen sind in [LS05] gesammelt worden. Da sie die oben beschriebene Situation verhindern, indem sie entsprechende Straffunktionen einführen, spricht man hier von auch penalized Likelihood.

Die allgemeine Form einer penalized Likelihood $l_q$ lautet wie folgt:

$$l_q(\boldsymbol{\theta}, \boldsymbol{\rho}, \boldsymbol{\gamma}) = l(\boldsymbol{\theta}, \boldsymbol{\rho}) - n \sum_{i=1}^{k} p_\lambda(\gamma_i) \tag{4.17}$$

Dabei ist $p_\lambda$ eine beliebige Straffunktion, $\lambda$ ein Regulierungsparameter und $\boldsymbol{\gamma} = (\theta_1, \ldots, \theta_k, \sigma^2)$. $L_1$ und $L_2$ sind einfache Straffunktionen, die die zu starke Gewichtung einzelner Parameter unterbinden. Dabei enthält $L_1$ einen linearen Strafterm

$$p_\lambda(\gamma) = \lambda \gamma \tag{4.18}$$

und $L_2$ einen quadratischen Strafterm:

$$p_\lambda(\gamma) = 0.5 \lambda \, |\gamma|^2 \tag{4.19}$$

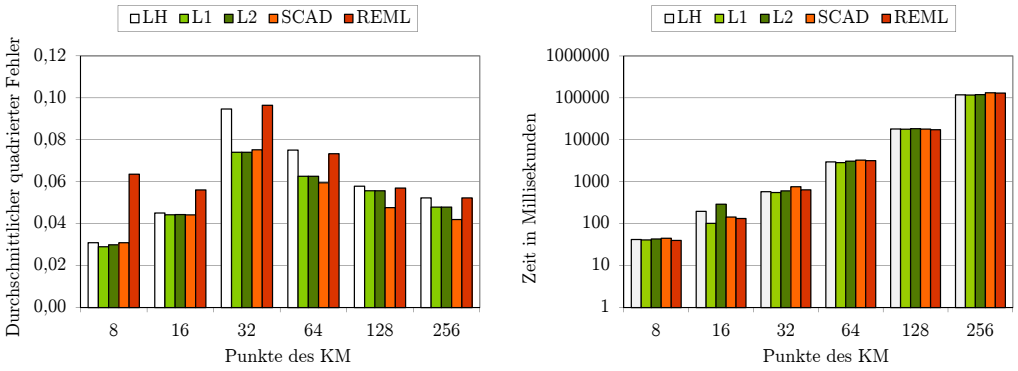



Abbildung 4.4.: Durchschnittlicher quadrierter Fehler und benötigte Zeit für unterschiedliche Likelihood-Funktionen für die Sinus-Funktion

Bei der Smoothly clipped absolute deviation (SCAD) handelt es sich um eine Straffunktion, die nach oben begrenzt ist. Dadurch sollen Randwerte zwar nicht überbewertet, auf der anderen Seite aber auch nicht übermäßig bestraft werden:

$$p_\lambda(\theta) = \lambda\left\{I(\theta \leq \lambda) + \frac{(a\lambda-\theta)_+}{(a-1)\lambda} I(\theta > \lambda)\right\}$$

$$\text{mit} \tag{4.20}$$

$$I(\theta) = -\frac{E\{l''(\theta)\}}{n}, a = 3,7$$

Ein weiterer Ansatz mit einer direkt modifizierten Likelihood-Funktion ist die Restricted Maximum Likelihood [PT71] (REML):

$$\begin{aligned} l_{reml}(\boldsymbol{\theta}, \boldsymbol{\rho}) = {} & \tfrac{n}{2} ln(2\pi) - \tfrac{n-1}{2} ln\sigma^2 - \tfrac{1}{2} ln\left|\boldsymbol{R}\right| - \tfrac{1}{2}\left|\mathbf{1}_n' \boldsymbol{R}^{-1} \mathbf{1}\right| \\ & -\tfrac{1}{2\sigma^2}\left(\boldsymbol{y} - \mathbf{1}_n\mu\right)' \boldsymbol{R}^{-1}\left(\boldsymbol{y} - \mathbf{1}_n\mu\right) \end{aligned} \tag{4.21}$$

Diese Auflistung erhebt keinen Anspruch auf Vollständigkeit, da Likelihood-Funktionen nicht nur im Bereich der KM genutzt werden und viele dieser Funktionen mit wenig Aufwand auf KM angewendet werden können. Im Folgenden werden verschiedene Likelihood-Funktionen für KM anhand der zweidimensionalen Funktionen Sinus, Ackley und Schaffer untersucht. Hierbei werden der durchschnittliche quadratische Fehler in der Approximation und die durchschnittlich benötigte Zeit betrachtet. Der Versuchsaufbau ist dabei wie folgt: Durch ein LHS werden die Punkte für das KM bestimmt. Eine Gitternetzsuche auf einem 121x121 Gitter mit 14621 Punkten wird durchgeführt, um das Maximum der Likelihood-Funktion zu finden. Dabei wird die für die Anpassung des KM benötigte Zeit gemessen. Anhand eines $11 \times 11$ Gitters wird der durchschnittliche quadratische Fehler des Metamodells gemessen. Die Anzahl der durch das LHS erstellten Punkte beträgt 4, 8, 16, ... bzw. 128. Die

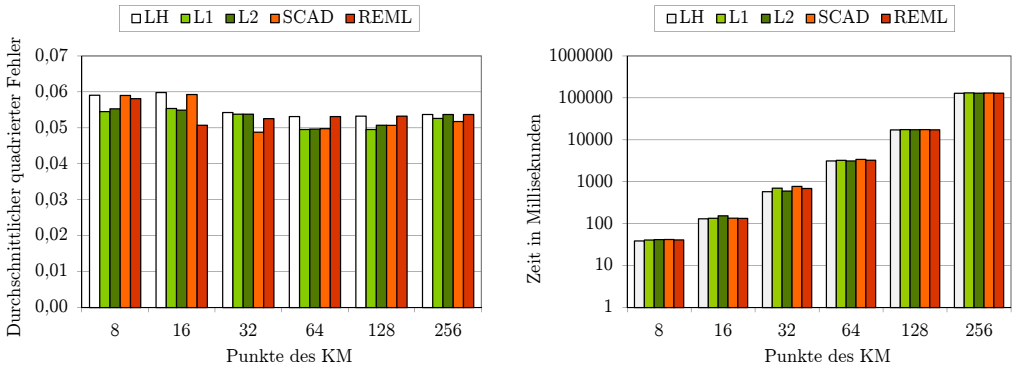



Abbildung 4.5.: Durchschnittlicher quadrierter Fehler und benötigte Zeit für unterschiedliche Likelihood-Funktionen für die Ackley-Funktion

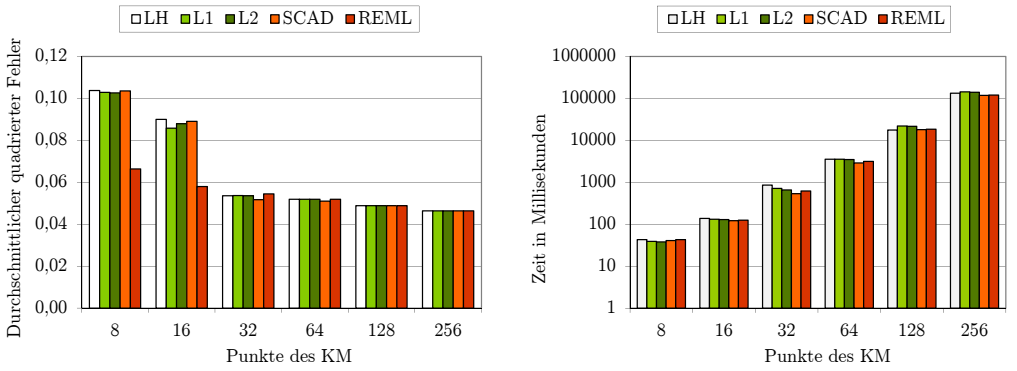



Abbildung 4.6.: Durchschnittlicher quadrierter Fehler und benötigte Zeit für unterschiedliche Likelihood-Funktionen für die Schaffer-Funktion

Messwerte werden über 100 Replikationen gemittelt. Betrachtet werden die normale Likelihood-Funktion (LH) in logarithmischer Form und die Methoden $L_1$, $L_2$, SCAD und REML aus Kapitel 4.3.2.

Die Ergebnisse der Versuchsreihen sind in den Abbildungen 4.4, 4.5 und 4.6 aufgeführt. Sie unterscheiden sich sowohl in den einzelnen Funktionen als auch zwischen den unterschiedlichen Größen der KM. Die Qualität der Ergebnisse hängt jedoch in erster Linie von den bekannten Punkten und nicht von der eingesetzten Likelihood-Funktion ab. Im Fall der Sinus-Funktion ist deutlich zu erkennen, dass LH und REML in der Regel zu schlechteren Anpassungen des KM als die anderen Verfahren führen, wobei SCAD tendenziell bessere Anpassungen ermöglicht. Wenn auch nicht so deutlich, zeigen die Ergebnisse der Experimente mit der Ackley-Funktion das gleiche Verhalten. Einzige Ausnahme bildet die Schaffer-Funktion, die bei wenigen Punkten (8 und 16 Punkte) durch REML deutlich besser angepasst wird. Sobald zusätzliche Punkte hinzugefügt werden, sind die Approximationen der restlichen Likelihood-Funktionen jedoch wieder gleichwertig. Da sich die Rechenzeit der Ver-

fahren nicht wesentlich unterscheidet, wird daher SCAD als Likelihood-Funktion in allen weiteren Experimenten eingesetzt.

### 4.3.3. Erweiterung der normalen Kriging Metamodelle

Kriging Metamodelle bieten viel Raum für Veränderungen ohne den ursprünglichen Charakter dieser Metamodelle zu verlieren. Neben den in den letzten Abschnitten vorgestellten Likelihood-Funktionen ist zum Beispiel die Korrelationsfunktion der KM ein Aspekt, der je nach Einsatzzweck variiert werden kann. In [Toc93] wird das **Dual Kriging** vorgeschlagen, dass durch eine entsprechend modifizierte Korrelationsfunktion sowohl globales als auch lokales Metamodell sein soll:

$$corr_{dual}(\boldsymbol{x}_1, \boldsymbol{x}_2) = \begin{cases} 1 - \left(\frac{d(\boldsymbol{x}_1,\boldsymbol{x}_2)}{d_{max}}\right)^3 & ; 0 \leq d(\boldsymbol{x}_1, \boldsymbol{x}_2) \leq d_{max} \\ 0 & ; d(\boldsymbol{x}_1, \boldsymbol{x}_2) > d_{max} \end{cases} \tag{4.22}$$

Entsprechend der normalen Korrelationsfunktion sinkt die Korrelation von Punkten mit deren Distanz. In normalen KM ist die Korrelation aller Punkte miteinander stets größer als 0. Im Ansatz des **Dual Kriging** haben jedoch Punkte, die weiter als $d_{max}$ von einander entfernt sind, keinen Einfluss aufeinander. Durch diese Veränderung soll das lokale Verhalten eines Metamodells besser nachgebildet werden können, ohne das globale Verhalten zu sehr zu vernachlässigen.

Ein Ansatz, der zwischen normalem und universellem Kriging anzusiedeln ist, wird **Limit Kriging** genannt [Jos05]. Durch einen neuen Schätzer für den Funktionswert soll in Fällen, bei denen die Annahme des konstanten Mittels der normalen KM nicht erfüllt ist, besser als der normale Schätzer sein. Darüber hinaus sollen empirische Untersuchungen gezeigt haben, dass dieser auch bei schlechter Anpassung des Metamodells robuster ist. Der Schätzer wird dabei wie folgt formuliert:

$$\begin{aligned} \hat{y}(x) =& \mu + \lambda(\boldsymbol{x})\boldsymbol{r}(\boldsymbol{x})'\boldsymbol{R}^{-1}(\boldsymbol{y} - \mathbf{1}_n\mu) \\ =& \mu\left(1 - \lambda(\boldsymbol{x})\boldsymbol{r}(\boldsymbol{x})'\boldsymbol{R}^{-1}\mathbf{1})\right) + \lambda(\boldsymbol{x})\boldsymbol{r}(\boldsymbol{x})'\boldsymbol{R}^{-1}\boldsymbol{y} \end{aligned} \tag{4.23}$$

Dabei wird angenommen, dass $\boldsymbol{r}(\boldsymbol{x})'\boldsymbol{R}^{-1}\mathbf{1}_n \neq 0$ gilt. Mit $\lambda(\boldsymbol{x}) = 1/\left(\boldsymbol{r}(\boldsymbol{x})'\boldsymbol{R}^{-1}\mathbf{1}_n\right)$ folgt daraus der folgende Schätzer:

$$\hat{y}(\boldsymbol{x}) = \frac{\lambda(\boldsymbol{x})\boldsymbol{r}(\boldsymbol{x})'\boldsymbol{R}^{-1}\boldsymbol{y}}{\lambda(\boldsymbol{x})\boldsymbol{r}(\boldsymbol{x})'\boldsymbol{R}^{-1}\mathbf{1}_n} \tag{4.24}$$

Ebenfalls zwischen normalem und universellem Kriging liegt das Verfahren aus [JHS08]. Dort wird ein neuer Schätzer vorgeschlagen, bei dem $\mu$ durch eine Menge von Funktionen ersetzt wird. Mit dieser Formulierung kommt dieser dem universellen Kriging sehr nah.

In [vBK03] werden **Detrendet KM** vorgeschlagen. Diese Form der KM lässt sich ebenfalls zwischen den normalen und den universellen KM einordnen. $\mu$ wird dabei durch ein lineares Regressionsmodell ersetzt. In diesem Artikel werden darüber hinaus lineare Regressionsmodelle, KM und **Detrendet KM** verglichen.

### 4.3.4. Kriging Metamodelle mit mehreren Quellen

In vielen Situation existieren neben der zu approximierenden Funktion $f$ weitere Funktionen bzw. Messdaten, die mit der Funktion $f$ korreliert sind. In diesem Abschnitt werden einige der in diesem Kontext entwickelten KM angerissen.

Co-Kriging [Cre93] (oder Cokriging) ist eine Variante der KM, mit der prozentuale Messwerte approximiert werden können. Dies ist z.B. bei der Zusammensetzung von Bodenproben notwendig. Bei der Analyse dieser Proben können in der Regel ohne zusätzlichen Zeitaufwand mehrere Bestandteile und deren Anteile an der Probe bestimmt werden. Dadurch erhält man nicht nur weitere Informationen über die Zusammensetzung, sondern kann weitere Informationen für die Approximation ableiten, da sich die Anteile zu 100% aufsummieren müssen.

In anderen Situation, wie z.B. Messung von Luft- bzw. Wasserwerten, sind nicht nur die gemessen Daten, sondern auch die Zeitpunkte der Messung von Bedeutung. Hier wurde zum Beispiel ein bayesisches hierarchische KM [vdKS06] entwickelt, dass die entsprechenden Bewegungen berücksichtigen kann. Bei einem anderen Ansatz wird ein Kriging Update Modell [KP99] vorgeschlagen, dass die Messungen dynamisch anpasst. Dabei wird universelles Kriging mit Kalman Filterung kombiniert.

In vielen Fällen wie z.B. in [BK06] existieren zu der zu betrachtenden Funktion $f$ einfacher zu bestimmende Approximationen $g$. In diesen Fällen ist es sinnvoll, beide Funktionen zu nutzen und die Anzahl der Auswertungen von $f$ gering zu halten bzw. das Ergebnis zu verbessern. Ein entsprechendes Verfahren wird z.B. in [HA05, Hua05] beschrieben. Weitere Varianten der KM wie das Gradientenkriging [Röt97] berücksichtigen und benötigen z.B. die Steigung an den Messpunkten.

# 5. Optimierung und Suchheuristiken

Unter der Optimierung einer Funktion $f$ versteht man die Bestimmung des Minimums (oder Maximums) dieser Funktion. Weder für beliebige noch für die hier betrachteten Funktionen existiert eine universelle Optimierungsmethode [Sch95], so dass für einzelne Problemstellungen speziell angepasste Verfahren entwickelt werden müssen. Dabei wird in dieser Arbeit ausschließlich die einkriterielle Optimierung betrachtet, wobei der Parametervektor sowohl reelle als auch ganzzahlige Variablen enthalten kann. Entsprechend der Annahme von Black-Box-Modellen aus Kapitel 2 ist bekannt, wie die Funktion für einzelne Punkte des Wertebereichs ausgewertet werden kann. Darüber hinaus ist zu beachten, dass die Modelle in der Regel stochastische Elemente enthalten. Zur Analyse stehen die in Kapitel 1.4 beschriebenen Verfahren mit ihren jeweiligen Vor- und Nachteilen zur Verfügung.

In der Praxis werden häufig Methoden eingesetzt, die einen erheblichen Arbeitsaufwand (z.B. verursacht durch manuelle Vereinfachung des Modells) bzw. signifikante Fehler verursachen können. Zu diesen Methoden gehört zum Beispiel die Linearisierung, bei der ein Modell soweit vereinfacht wird, dass es sich als lineares Optimierungsproblem formulieren lässt, welches dann mittels Simplexverfahren gelöst wird. Die in den vorherigen Kapiteln getroffenen Annahmen lassen eine solche Vereinfachung jedoch nicht zu. Daher müssen andere Verfahren entwickelt bzw. verbessert werden, insbesondere wenn die Auswertung der Zielfunktion zeitaufwändig ist.

In dem Bereich der Optimierung aufwändiger Zielfunktionen haben sich in den letzten Jahren aus den etablierten Verfahren z.B. Kriging Metamodelle und die darauf basierten Optimierungsverfahren hervorgehoben. In dieser Arbeit soll daher der Schwerpunkt auf den Kriging Metamodellen und der Kriging Metamodell basierten Optimierung liegen. Die zu optimierenden Funktionen sind dabei durch ereignisdiskrete Modelle gegeben.

In diesem Kapitel wird auf die Themen Optimierung und Suchheuristiken eingegangen. Da es sich um ein breites Feld mit zum Teil unterschiedlichen Begriffsdefinitionen handelt, werden zuerst der untersuchte Bereich eingegrenzt und die benötigten Definitionen festgelegt. Anschließend werden einige bekannte Verfahren sowie einige Modifikationen dieser Verfahren vorgestellt.

## 5.1. Bereichseingrenzung und Begriffsdefinition

Der Begriff der Optimierung hat nicht nur in der Informatik eine lange Tradition. Aus mathematischer Sicht ist eine Optimierung die Bestimmung eines Optimums (Minimums oder Maximums). Da die Bestimmung des Maximums einer Funktion $f$ der Minimierung der Funktion $-f$ entspricht, wird im weiteren nur die Minimierung betrachtet:

**Definition 14 (Optimierung)** *Eine **Optimierung** ist die Bestimmung der Position eines Minimums $y^*$ einer Funktion $f$ mit $k$ Faktoren: $y^* = argmin\{f(y)|y \in \mathbb{R}^k\}$.*

Für die bisher beschriebenen Funktionen kann jedoch nicht garantiert werden, dass dieses Optimum gefunden wird. Da die betrachteten Zielfunktionen (siehe Kapitel 2) nicht genauer bekannt sind, kann das Auffinden eines globalen Optimums nur durch vollständige Untersuchung des Definitionsbereichs der Zielfunktion erreicht werden. Da der Definitionsbereich in der Regel unendlich viele Werte enthält, kann dadurch in endlicher Zeit das Auffinden des globalen Optimums nicht garantiert werden. Es existieren jedoch trotz dieser Einschränkung verschiedene Verfahren, die unterschiedliche Strategien zur Suche nach einem Optimum verfolgen.

In der Regel ist der Wertebereich von $f$ beschränkt. Diese Beschränkungen sind im Allgemeinen als Ungleichungen gegeben, die Nebenbedingungen genannt werden. Diese wird als Optimierung unter Nebenbedingungen oder als beschränkte Optimierung bezeichnet. Eine Optimierung ohne Beschränkung des Wertebereichs wird zur Abgrenzung als unbeschränkte Optimierung bezeichnet.

**Definition 15 (Optimierung unter Nebenbedingungen)** *Eine **Optimierung unter Nebenbedingungen** ist die Bestimmung eines Minimums $y^*$ einer Funktion $f$ bei Einhaltung einer Menge von $m$ Nebenbedingungen, die in der Form $n_i(y) \leq b_i$ mit $i = 1, \ldots, m$ gegeben sind:*

$$\begin{aligned} y^* &= argmin\left\{f(y)|y \in \mathbb{R}^k\right\} \\ &\text{u.d.N.} \\ \forall i \in \{1, \ldots, m\} &: n_i(y) \leq b_i \end{aligned} \tag{5.1}$$

Bei den später betrachteten Verfahren ist der Wertebereich $W$ durch einen Hyperkubus eingeschränkt. Weitere Nebenbedingungen können darüber hinaus hinzugefügt werden (siehe Abschnitt 5.2). Die Optimierung mit Nebenbedingungen kann dementsprechend wie folgt beschrieben werden:

$$y^* = argmin\,\{f(y)|y \in W\}$$

$$\text{u.d.N.} \tag{5.2}$$

$$\forall i \in \{1, \ldots, m\} : n_i(y) \leq b_i$$

Des weiteren existiert die lokale Optimierung, die gegenüber der globalen Optimierung nur das Auffinden eines lokalen Minimums garantiert. Eine Optimierung ohne diese Einschränkung wird als globale Optimierung bezeichnet. Es ist zu beachten, dass die Ergebnisse der lokalen Optimierung im Fall von Funktionen mit genau einem Minimum den Ergebnissen einer globalen Optimierung entsprechen. Bei Verfahren, die eine lokale Optimierung durchführen, ist das gefundene Optimum in der Regel von dem Verlauf der zugrunde liegenden Funktion und der Konfiguration des Verfahrens abhängig. Mehrere Durchführungen des gleichen Verfahrens auf der gleichen Funktion können dadurch zu unterschiedlichen Ergebnissen führen.

Des weiteren können die beiden grundlegenden Bereiche einkriterielle Optimierung, bei der die Anzahl der Zielfunktionen durch 1 beschränkt ist, und multikriterielle Optimierung mit mehreren Zielfunktionen unterschieden werden. Zur Lösung von multikriteriellen Optimierungsproblemen existiert eine Reihe von Ansätzen. Zu diesen gehören z.B. die Rückführung auf eine einkriterielle Optimierung mittels Gewichtungsfunktionen und die Untersuchung von Pareto Fronten [Wei07]. Da der Fokus dieser Arbeit auf einkriteriellen Optimierungsproblemen liegt, sollen diese nun genauer betrachtet werden.

**Definition 16 (Einkriterielle Optimierung)** ***Einkriterielle Optimierung*** *ist eine Optimierung, deren Optimierungsziel durch die Minimierung bzw. Maximierung genau einer Zielfunktion festgelegt ist.*

Da für die betrachteten Funktionen nicht garantiert werden kann, das globale Optimum zu finden, haben sich verschiedene Suchheuristiken entwickelt, die auf bestimmte Arten von Funktionen ausgerichtet sind. Diese Suchheuristiken lassen sich in lokale und globale Verfahren unterteilen. Bei lokalen Verfahren wird angenommen, dass es sich bei der Zielfunktion $f$ um eine unimodale Funktion handelt. Durch diese Annahme kann i.d.R. die Anzahl der Auswertungen der Funktion $f$ gegenüber globalen Verfahren deutlich reduziert werden. Grundsätzlich sind diese Verfahren zwar auf unimodale Funktionen ausgerichtet, es lässt sich jedoch empirisch zeigen, dass einige dieser Verfahren auch für bestimmte multimodale Funktionen geeignet sind. Globale Verfahren sind darauf ausgelegt mit hoher Wahrscheinlichkeit globale Optima zu erfassen. Dabei besteht im Allgemeinen ein Tradeoff zwischen der Wahrscheinlichkeit ein globales Optimum zu finden und der Anzahl der benötigten Auswertungen.

Grundsätzlich lässt sich sagen, dass es kein Verfahren gibt, dass in allen Fällen allen anderen Verfahren überlegen ist [WM97]. Betrachtet man Verfahren, die auf

unimodale Funktionen beschränkt sind, können diese von einem Punkt ausgehend direkt in die Richtung des globalen Optimums gehen. Eine Betrachtung anderer Regionen kann aufgrund der Unimodalität der Zielfunktion entfallen. Verfahren für multimodale Funktionen müssen jedoch auch bei unimodalen Funktionen zuerst den Wertebereich untersuchen, um das Verhalten der Funktion zu analysieren. Daraus resultiert eine erhöhte Anzahl an betrachteten Punkten. Auf der anderen Seite erreichen Verfahren für unimodale Funktionen auf multimodalen Funktionen in der Regel nur lokale Optima, wobei das Ergebnis vom gewählten Startpunkt abhängt. Dieses Risiko wird bei Verfahren für Multimodale Funktion reduziert, indem eine weiter gefächerte Suche durchgeführt wird. Zuletzt wird noch auf die Optimierung der Funktion $f_{\mathbf{a}}$ aus Kapitel 2.1 verwiesen, bei der alle systematischen Suchen das Optimum nur in Ausnahmefällen finden können.

## 5.2. Diskrete Parameter und Nebenbedingungen

In dieser Arbeit liegt der Schwerpunkt der Betrachtungen auf Funktionen mit kontinuierlichen Parametern. In vielen Fällen treten jedoch auch diskrete Parameter auf (wie z.B. im Lagerhaltungssystem aus Abschnitt 2.2.2). Daher wird an dieser Stelle kurz darauf eingegangen, in welchen Fällen diskrete Parameter trotzdem ohne tief greifende Änderungen verarbeitet werden können. Grundsätzlich können drei Arten von diskreten Parametern auftreten:

1. Zwischen den Zahlenwerten des Parameters besteht keine direkte Verbindung.
2. Zwischen den Zahlenwerten des Parameters besteht eine direkte Verbindung:
   a) Die Anzahl der Werte des Parameters ist klein.[2]
   b) Die Anzahl der Werte des Parameters ist nicht klein[2].

Für den ersten Fall müssen entweder entsprechende Verfahren entwickelt werden, die diese Besonderheit berücksichtigen. Die hier vorgestellten Verfahren setzen jedoch eine direkte (mathematische) Beziehung zwischen den Parameterwerten voraus, die in diesem Fall nicht gegeben ist. Solange die Anzahl der Parameterwerte klein ist, wird empfohlen für jeden Parameterwert eine eigene Optimierung durchzuführen und anschließend das Optimum mit dem besten Wert zu wählen [MM02]. Analog kann dies auf den Fall 2a angewendet werden. Im Fall 2b verhält sich der Parameter bereits ähnlich wie Parameter mit kontinuierlichem Wertebereich. Daher genügt es, den Wert lediglich auf den nächsten gültigen Wert zu runden. Hierbei ist jedoch zu beachten, dass korrektes Runden NP-schwer ist.

[2]Die Bestimmung der Grenze der Anzahl der Werte eines Parameters, so dass diese als klein bzw. nicht klein gelten, hängt von verschiedenen Faktoren wie z.B. dem Modell, der benötigten Zeit für eine Auswertung des Modells und dem Einfluss des Parameters ab. Eine detaillierte Untersuchung dieses Themas steht jedoch nicht im Fokus dieser Arbeit.

Darüber hinaus können bei der Optimierung ereignisdiskreter Modelle zwei Arten von Nebenbedingungen auftreten. Die Nebenbedingungen der ersten Art hängen ausschließlich von Parametern des Modells ab. Die Nebenbedingungen zweiter Art berücksichtigen auch Ergebnisse der Auswertung des Modells. Der wichtigste Unterschied der beiden Arten besteht darin, dass die erste Art bereits vor der Auswertung des Modells überprüft werden kann, die zweite Art jedoch nicht. Da die Auswertungen in der Regel teuer sind, ist es nicht sinnvoll, die Ergebnisse dieser Auswertung zu verwerfen, wenn festgestellt wird, dass eine Nebenbedingung verletzt ist.

Daher wird die Einhaltung der Nebenbedingungen der ersten Art bereits vor der Auswertung sichergestellt. Beim Einsatz ausschließlich linearer Nebenbedingungen kann der zu untersuchende Punkt so verschoben werden, dass die Nebenbedingungen erfüllt sind. Für die Verschiebung wird jeweils eine verletzte Nebenbedingung genutzt, zu der eine Senkrechte berechnet wird. Der Punkt wird entlang dieser Senkrechten verschoben bis er die Nebenbedingung genau erfüllt. Die Überprüfung der diskreten Parameter wird dabei vorübergehend nicht durchgeführt. Wurde ein gültiger Punkt gefunden, werden nun die diskreten Parameter gerundet. Entsteht dadurch ein Punkt, der eine Nebenbedingung verletzt, wird unter den diskreten Parametern eine Nachbarschaftssuche durchgeführt, bis der nächstgelegene Punkt gefunden wurde, der keine Nebenbedingung verletzt. Analog können nichtlineare Nebenbedingungen unterstützt werden, indem der die Nebenbedingungen verletzende Punkt in Richtung des nächstgelegen bekannten und gültigen Punkts verschoben wird. Wenn diese Anpassung der Punkte für die nachfolgend beschriebenen Suchheuristiken transparent durchgeführt wird, müssen diese Situationen bei der Konzeption und Umsetzung der Algorithmen nicht weiter betrachtet werden.

Auf die Einhaltung der Nebenbedingungen der zweiten Art wird in dieser Arbeit nicht genauer eingegangen und bei den später vorgestellten Verfahren nicht betrachtet. Diese Nebenbedingungen können jedoch für jedes Modell durch eine geeignete Straf- oder Barrierefunktion [BSMM97] umgesetzt werden.

Eine Straffunktion muss dabei folgende Voraussetzungen erfüllen:

1. Die Straffunktion ist 0, wenn keine Nebenbedingung verletzt wurde.
2. Die Straffunktion ist stetig.
3. Je stärker die Nebenbedingungen verletzt sind, desto größer ist der Wert der Straffunktion.
4. Die Straffunktion muss im Verhältnis zur Zielfunktion ausreichend stark ausgeprägt sein, so dass sich die Werte der kombinierten Funktion in Richtung des gültigen Bereichs verbessern.

Bedingung 1 ist trivial, da im zulässigen Bereich die Zielfunktion optimiert werden soll. Die Bedingungen 2, 3 und 4 stellen sicher, dass die Suchheuristiken in den gültigen Bereich geleitet werden. Zur Einhaltung von Bedingung 4 wird in der Regel

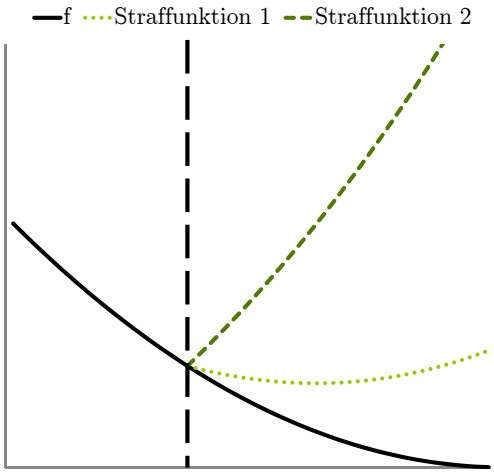



Abbildung 5.1.: Zielfunktion $f$ mit Grenze des zulässigen Bereichs (gestrichelte schwarze Linie) und zwei Straffunktionen außerhalb des zulässigen Bereichs

ein iterativer Ansatz gewählt, so dass der Einfluss der Straffunktion zu Beginn einer Optimierung gering ist und im Laufe der Optimierung ansteigt. In Abbildung 5.1 ist links eine Funktion $f$ abgebildet, die im nicht zulässigen Bereich bessere Funktionswerte als im zulässigen Bereich enthält. Durch Straffunktion 1 wird das Verhalten abgeschwächt, das Minimum liegt jedoch weiterhin im nicht zulässigen Bereich. Durch Straffunktion 2 wird das Minimum genau auf die Grenze des zulässigen Bereichs verschoben. Im Allgemeinen kann jedoch durch Straffunktionen nicht garantiert werden, dass das Minimum im zulässigen Bereich liegt.

Barrierefunktionen verhindern das Verlassen des zulässigen Bereichs, indem sie bereits im zulässigen Bereich einen Strafterm der Zielfunktion hinzufügen. Dieser ist abhängig von der Nähe zum nicht zulässigen Bereich. Dabei ist der Strafterm an der Grenze zum zulässigen Bereich gleich $\infty$. Eine Funktion $f$ mit zwei Barrierefunktionen ist rechts in Abbildung 5.2 dargestellt. Mit beiden Barrierefunktionen ist es nicht möglich den optimalen Punkt an der Grenze zum unzulässigen Bereich zu erreichen, Barrierefunktion 1 modifiziert die Funktion $f$ jedoch stärker und erhöht damit die Distanz zum Optimum stärker.

## 5.3. Lokale und globale Verfahren

Es existiert eine Reihe von Verfahren, die zur Optimierung genutzt werden. Je nach Einsatzgebiet haben sich dabei unterschiedliche Verfahren etabliert. Ein Überblick

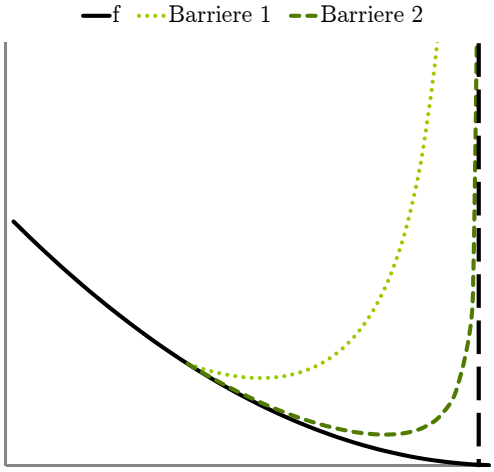



Abbildung 5.2.: Zielfunktion $f$ mit Grenze des zulässigen Bereichs (gestrichelte schwarze Linie) und zwei Barrierefunktionen innerhalb zulässigen Bereich

über dieses Thema ist zum Beispiel in [Sch95] zu finden. In den folgenden Abschnitten sollen daher nur Verfahren beschrieben werden, die im weiteren Verlauf der Arbeit genutzt werden. Zu diesen gehören die lokalen Verfahren **Response Surface Methodology** (RSM) und **Pattern Search** (PS) sowie die globalen Verfahren **Evolutionäre Algorithmen** (EA) und **Kriging Metamodell basierte Optimierung** (KMO). Diese und einige weitere Verfahren sind in das Tool OPEDo [BMKT06] integriert. Auch die im späteren Teil dieser Arbeit vorgestellten Erweiterungen sind dort verfügbar. Da auf Kriging Metamodelle und die darauf basierende Optimierung in dieser Arbeit ein stärkerer Fokus gelegt wird, wird dieses Verfahren in einem eigenen Kapitel beschrieben und hier zunächst ausgespart.

Neben den bisherigen Unterteilungen kann man die Verfahren auch anhand ihrer Ansätze unterscheiden. PS gehört zu den direkten Verfahren [LTT00]. Diese Verfahren wählen anhand eines vorgegebenen Schemas neue Punkte zur Auswertung aus. Ist einer der neuen Punkte besser als die bisher bekannten, ist dieser das aktuelle (Zwischen-) Ergebnis. Diese Art von Verfahren nutzt nur in sehr geringem Ausmaß die Informationen, die durch bereits bekannte Punkte ermittelt wurden. Meistens wird nur der bisher beste bekannte Punkt zur Berechnung neuer Punkte verwendet. Weitere Punkte werden durch Metamodell-basierte Verfahren wie RSM und KMO genutzt. Diese Verfahren treffen entsprechend einer festgelegten Strategie ihre Entscheidung auf einer Menge von Punkten, wobei nicht zwingend alle bekannten Punkte berücksichtigt werden. Des weiteren existieren biologisch inspirierte Verfahren. Dazu werden von der Natur entwickelte Verhaltensweisen analysiert und vollständig oder teilweise als Algorithmen formuliert. Das populärste Beispiel für diesen Bereich sind die EA, die von einem Generationenmodell ausgehen und bei denen die Eigen-

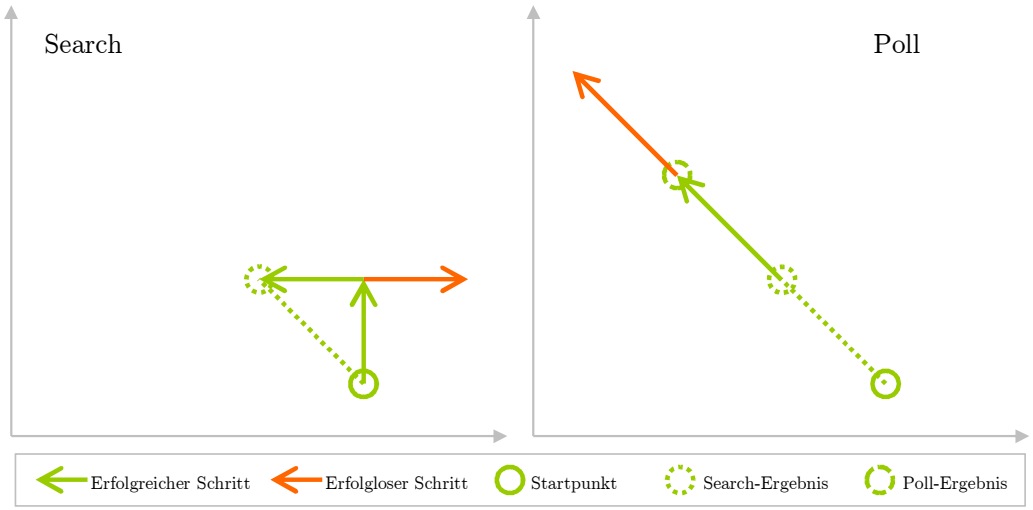



Abbildung 5.3.: Beispiele für das Verhalten von Pattern Search in den Schritten Search und Poll

schaften der Nachfahren von ihren Eltern abhängen, die Nachfahren aber auch in Grenzen eigene Ausprägungen einiger Eigenschaften entwickeln können.

Die allgemeine Fassung dieser Verfahren bezieht sich in der Regel auf deterministische Funktionen. Die hier aufgeführten Beschreibungen berücksichtigen zuerst nur diese grundlegende Fassung. Spezielle Anpassungen auf bestimmte Situationen werden jeweils direkt im Anschluss dargestellt. Allgemeine Methoden zur Anpassung auf stochastische Funktionen, die für die verschiedenen Verfahren genutzt werden können, werden in einem späteren Kapitel beschrieben. Die Vorstellung der EA wird im Gegensatz zu RSM, PS und KMO nur kurz ausfallen und nicht auf Besonderheiten für ereignisdiskrete Modelle eingehen, da EA im Rahmen dieser Arbeit nur als Hilfsmittel betrachtet wird, um die für KMO benötigten KM zu erstellen.

## 5.4. Pattern Search

Pattern Search (PS) gehört zu den direkten Suchverfahren (DS). DS sind iterative Verfahren, bei denen jede Iteration in die zwei Schritte Search und Poll aufgeteilt ist. Im Search-Schritt wird eine Menge von Kandidatenpunkten (genannt Mesh) um den bisher besten bekannten Punkt erstellt. Der Mesh besteht dabei meistens aus den Eckpunkten eines Gitters mit $l \geq 3$ Schnittpunkten für jede Dimension des Suchraums. Da die Größe des Mesh mit der Anzahl der Dimensionen exponentiell steigt, werden die Punkte des Mesh in der Regel nicht explizit berechnet, sondern

nur eine Methode zu deren Bestimmung festgelegt. Aus dem Mesh werden anschließend Punkte ausgewählt und untersucht. Im Poll-Schritt wird aus der Differenz des besten bisher gefundenen Punkts und dem Startpunkt der Iteration die Richtung für die weitere Suche bestimmt. Die Koordinaten der Punkte werden solange um diese Differenz erhöht, bis sich der Funktionswert nicht weiter verbessert. Der beste gefundene Punkt wird anschließend Startpunkt der nächsten Iteration. Neben der Beschreibung dieses Verfahrens existieren - wie für viele weitere DS - auch Untersuchungen zum Konvergenzverhalten [Tor97].

Zu PS wurden verschiedene Varianten wie zum Beispiel **Generalized Pattern Search**[Tor97] entwickelt. Die folgende Beschreibung orientiert sich an [Sch95]. Eine Darstellung als Pseudocode befindet sich unter Algorithmus 5.1 und eine grafische Darstellung der Schritte in Abbildung 5.3.

PS betrachtet für die Generierung des Mesh einen lokalen Bereich mit Halbbreite $w$, die nach jeder Iteration verkleinert werden kann, um den Startpunkt $x$ der Iteration. Der initiale Wert von $w$ entspricht in der eingesetzen Implementierung 40% des Wertebereichs jeder Dimension. Der Mesh entspricht der Menge $\{y \in \mathbb{R}^k | \forall i = 1, \ldots, k : y_i \in \{x_i - w, x_i, x_i + w\}\}$. Für jede Dimension des Suchbereichs existiert ein $k$-Vektor $s_i$ mit $i = 1, \ldots, k$, der die Richtung des letzten Sucherfolgs für diese Dimension enthält. Initial sind die Werte der Vektoren wie folgt festgelegt:

$$s_{i,j} = \begin{cases} 1 & : \quad j = i \\ 0 & : \quad \text{sonst} \end{cases} \qquad \text{mit} \quad j = 1, \ldots, k \tag{5.3}$$

Zu Beginn des Mesh-Schritts wird der Punkt $x_{best} = x$ gewählt. Für jede Dimension $i$ des Suchbereichs werden nacheinander folgende Schritte durchgeführt. Zuerst werden die Punkte $x_{next,1} = x_{best} + w \cdot s_i$ und $x_{next,2} = x_{best} - w \cdot s_i$ berechnet. Ist $f(x_{next,1}) < f(x_{best})$ so wird $x_{best} := x_{next,1}$ und mit der nächsten Dimension fortgefahren. Sonst wird wird überprüft, ob $f(x_{next,2})$ kleiner als $f(x_{best})$ ist. Ist dies der Fall wird $x_{best}$ gleich $x_{next,2}$, $s_i$ auf den Wert $-s_i$ geändert und mit der nächsten Dimension fortgefahren. Sind alle Dimensionen abgearbeitet, wird mit dem Poll-Schritt aus DS fortgefahren.

War ein Optimierungsschritt erfolglos, wird angenommen, dass der aktuelle lokale Bereich das Optimum enthält. Da die untersuchten Punkte mindestens $w$ vom besten Punkt entfernt liegen, erscheint es sinnvoll die Halbbreite des lokalen Bereichs zu verkleinern, um sich dem Optimum weiter annähern zu können. Die neue Breite des lokalen Bereichs wird in der Regel als $w/2$ festgelegt.

## 5.5. Response Surface Methodology

RSM ist eine Sammlung statistischer und mathematischer Methoden, die zur Optimierung von Systemen genutzt werden können. Dieses Verfahren wurde ausführ-

```
1: Bestimme Startpunkt best, Fehlergrenze w_min und Startweite w_start;
2: Initialisiere k-Vektor s_i mit 0_k;
3: for i ∈ {1,...,k} do
4:    s_{i,i} = w_start;
5: end for
6: last = best;
7: step = best;
8: while ∃j : |s_j| > w_min do
9:    for i ∈ {1,...,k} do
10:      current = step + s_i;
11:      if messwert(current) < messwert(step) then
12:         current = step - s_i;
13:      end if
14:      if messwert(current) > messwert(step) then
15:         step = current;
16:      end if
17:   end for
18:   if messwert(step) > messwert(best) then
19:      d = step - best;
20:      if |d_i| ≠ 0 then
21:         for i ∈ {1,...,k} do
22:            s_i = s_i * sign(s_{i,i}) * sign(d_i);
23:         end for
24:      end if
25:      repeat
26:         current = step + d;
27:         best = step;
28:         step = current;
29:      until messwert(step) < messwert(best)
30:   else
31:      for i ∈ {1,...,k} do
32:         s_i = s_i / 2;
33:      end for
34:   end if
35: end while
36: Gib best als besten gefundenen Punkt zurück;
```

**Algorithmus 5.1:** Pattern Search Pseudocode

lich in [MM02] und in Hinblick auf Simulation in [NvOPD00] diskutiert. Die hier aufgeführte Beschreibung orientiert sich im Wesentlichen an diesen Ausführungen. Die allgemeine Beschreibung richtet den Fokus auf eine manuelle Durchführung des Verfahrens. Der folgende Text beschreibt ein vollständig automatisiertes Verfahren, bei dem manuelle Entscheidungsmöglichkeiten durch computerbasierte Bewertungs- und Entscheidungsverfahren ersetzt wurden. Weitere Entwicklungen, wie sie zum Beispiel in [DKM08] zur weiteren Anpassung an die simulationsgestützte Optimierung beschrieben sind, zeigen, dass in diesem Bereich immer noch Forschungsbedarf und -interesse besteht.

RSM ist ein iteratives Verfahren, das grundsätzlich in zwei Phasen eingeteilt werden. In der ersten Phase wird versucht, in die Nähe eines Optimums zu gelangen. In der zweiten Phase soll das Optimum genauer bestimmt werden. Das Verfahren startet an einem von außen vorgegebenen Punkt. Dieser Punkt ist der initiale beste Punkt, wobei als bester Punkt jeweils der Punkt bezeichnet wird, der den besten Funktionswert bzgl. des Optimierungsziels besitzt. In jedem Schritt der ersten Phase werden zuerst einige Punkte in der Nähe betrachtet und eine Hyperebene berechnet, die möglichst nah an diesen Punkten und dem besten Punkt liegt. Der Richtung des steilsten Abstiegs dieser Hyperebene wird ausgehend vom aktuell besten Punkt gefolgt, um bessere Punkte zu finden. Wenn keine besseren Punkte mehr gefunden werden, wird der Schritt beendet. In der zweiten Phase wird davon ausgegangen, dass sich das Verfahren bereits dem Optimum angenähert hat. Hier ist die Wahrscheinlichkeit sehr gering, dass die Umgebung des aktuell besten Punkts gut durch eine Hyperebene anzupassen ist. Daher wird versucht die Umgebung durch eine quadratische Funktion zu approximieren, zu der das Optimum berechnet wird. Anschließend wird der so errechnete Punkt mit dem bisher besten Punkt verglichen und entweder als neuer bester Punkt übernommen oder verworfen. Analog zu PS wird im Laufe des Verfahrens der betrachtete Bereich verkleinert.

### 5.5.1. Vorbereitungen

RSM betrachtet als lokales Suchverfahren jeweils nur einen Teil des gültigen Wertebereichs. Mittelpunkt dieses Bereichs ist der aktuell beste Punkt, um den sich ein Hyperkubus befindet, der den aktuell betrachteten Wertebereich darstellt. Um einige Berechnungen innerhalb des Verfahrens zu vereinfachen, wird der sogenannte kodierte Wertebereich eingeführt. Dabei gilt:

1. Der beste Punkt ist der Nullpunkt.
2. Bei allen Eckpunkten $x_i$ des Hyberkubus gilt deren Parameterwerte: $x_{i,j} \in \{-1; 1\}$

Der ursprüngliche Wertebereich wird zur Unterscheidung natürlicher Wertebereich genannt.

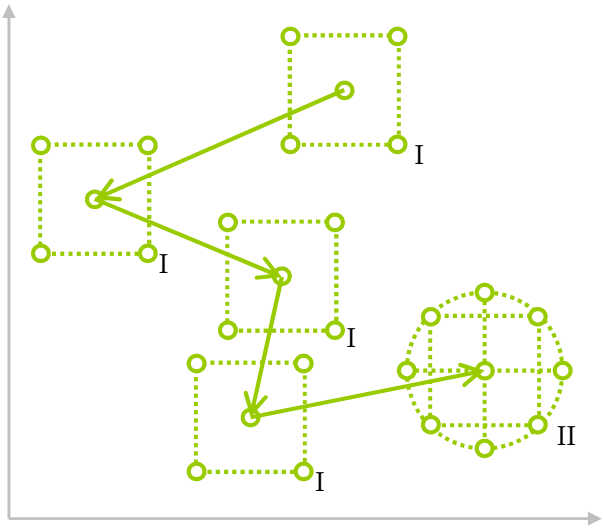



Abbildung 5.4.: Ablauf der Response Surface Methodology mit faktoriellen Designs (I) und zentral zusammengesetzten Designs (II)

Durch diese Transformation soll zusätzlich gewährleistet werden, dass der Wertebereich jedes Parameters gleichmäßig untersucht wird. Um diesen Effekt zu erreichen wird die Halbbreite des Suchbereichs $w$ als relative Größe angegeben. Wenn der Wertebereich des $i$-ten Parameters durch $[x_{min,i}; x_{max,i}]$ gegeben ist, ist die effektive Halbbreite des Suchbereichs für diesen Parameter $b_i = w \cdot (x_{max,i} - x_{min,i})$. Mittels des aktuell besten Punkts $c$ lassen sich die Grenzen des Suchbereichs für den $j$-ten Parameter als untere Grenze $l_i = c_i - b_i$ und obere Grenze $u_i = c_i + b_i$ berechnen. Für einen Wert $x_i$ aus dem natürlichen Wertebereich des $i$-ten Parameters ergibt sich der Wert $x'_i$ im kodierten Wertebereich wie folgt:

$$x'_i = \frac{x_i - c_i}{b_i} \tag{5.4}$$

Im weiteren Verlauf dieses Texts werden ausschließlich kodierte Variablen betrachtet.

### 5.5.2. Ablauf der Optimierung und Einsatz der dabei genutzten Methoden

Wie bereits erwähnt, wird die Optimierung mittels RSM iterativ und in zwei Phasen durchgeführt. Zuerst werden nun die Gemeinsamkeiten der Optimierungsschritte beider Phasen beschrieben. Auf die Unterschiede wird in späteren Abschnitten eingegangen. RSM führt die Optimierung Metamodell-basiert durch. Dabei wird zu

Beginn jedes Schritts ein lineares Regressionsmodell erstellt. Dieses Regressionsmodell wird mittels der Methode der kleinsten Fehlerquadrate an einige Punkte im lokalen Bereich angepasst. Diese Designpunkte werden dabei durch ein geeignetes Experimentdesign (siehe Kapitel 3) ausgewählt. Das Metamodell muss anschließend ggf. noch auf seine Anpassungsüte überprüft werden, worauf in einem der folgenden Abschnitte detaillierter eingegangen wird. Auf der Basis dieses Regressionsmodells werden viel versprechende Punkte auswählt, ausgewertet und überprüft, ob ihre Messwerte besser als diejenigen des aktuell besten Punkts sind. Das Ergebnis eines Optimierungsschritts ist entweder der alte oder ein neuer bester Punkt mit besserem Messwert. Auf diese Weise nähert man sich einem lokalen Optimum an (siehe Abbildung 5.4).

War ein Optimierungsschritt erfolglos, wird angenommen, dass der aktuelle Suchbereich das Optimum enthält. Da die Designpunkte mindestens $w$ vom besten Punkt entfernt liegen, erscheint es sinnvoll den Suchbereich zu verkleinern, um sich dem Optimum weiter annähern zu können. Die neue Halbbreite des Suchbereichs $w'$ wird in der Regel als $w/2$ festgelegt.

Nach jedem Optimierungsschritt muss der Algorithmus entscheiden, wie weiter vorzugehen ist. Dabei kann ohne Veränderung fortgefahren, die Phase gewechselt oder abgebrochen werden. Die zwei Hauptkriterien für diese Entscheidung „Größe des Suchbereichs“ und „Anzahl der Auswertungen bzw. Optimierungsschritte“ sollten dabei auf jeden Fall berücksichtigt werden. Ist der Suchbereich ausreichend klein, ist erstens keine wesentliche Verbesserung des Messwerts zu erwarten und zweitens haben auch als kontinuierlich angenommene Faktoren eine beschränkte Genauigkeit. Über die Anzahl der Auswertungen bzw. Optimierungsschritte kann der Zeitaufwand für die Optimierung abgeschätzt werden. Diese beiden Bedingungen eignen sich einerseits als Abbruchkriterium andererseits auch als Entscheidungsmerkmal für den Einsatz eines Modells zweiter Ordnung. Modelle zweiter Ordnung sollten aufgrund der hohen Anzahl benötigter Auswertungen nicht von Beginn an genutzt werden. Gegen Ende der Optimierung befindet man sich jedoch meist in der Nähe eines Optimums. Ein solcher Bereich lässt sich durch ein lineares Regressionsmodell zweiter Ordnung deutlich besser approximieren als durch ein lineares Regressionsmodell erster Ordnung. Daher ist hier der erhöhte Aufwand gerechtfertigt.

### 5.5.3. Phase 1

In der ersten Phase wird die Optimierung basierend auf einem linearen Regressionsmodell erster Ordnung (siehe Kapitel 4) durchgeführt, das auf der Basis ermittelter Messwerte eine Hyperebene approximiert. Die Punkte, für die die entsprechenden Messwerte bestimmt werden, werden durch ein faktorielles oder teilfaktorielles Design (siehe Kapitel 3) bestimmt und entsprechen allen Eckpunkten (oder im Falle des teilfaktoriellen Designs einem Teil der Eckpunkte) des lokalen Suchbereichs.

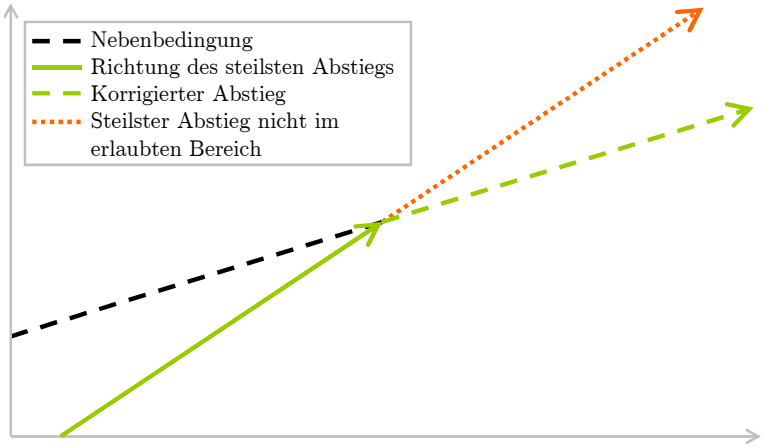



Abbildung 5.5.: Korrektur der Richtung des steilsten Abstiegs

In Einzelfällen müssen diese Designs durch Computer Generierte Designs ersetzt werden (siehe Kapitel 3).

Mittels der Senkrechten $\beta$ des Regressionsmodells kann die Richtung des steilsten Abstiegs berechnet werden. Zur Bestimmung der Schrittweite können verschiedene Heuristiken angewendet werden. Die empfohlene Methode wählt die Schrittweite konstant, so dass der erste Schritt auf eine der Seiten des Hyperkubus führt, der durch die Punkte eines Designs aufgespannt wird. Potentielle neue beste Punkte ergeben sich nur als Summe des aktuell besten Punkts und des Richtungsvektors $\bar{\beta} = \beta / max\,(\beta_1, \ldots, \beta_k)$.

Ist der Messwert des potentiellen neuen besten Punkts besser als der des Alten, wird der neue Punkt als neuer bester Punkt akzeptiert und die schrittweise Suche fortgeführt; andernfalls wird die Suche beendet. Bei dieser Vorgehensweise ist zu beachten, dass die Nebenbedingungen der zu optimierenden Funktion eingehalten werden. Dazu muss ggf. der Richtungsvektor korrigiert oder der Optimierungsschritt beendet werden. Eine detaillierte Beschreibung von Verfahren zur Korrektur des Richtungsvektors findet sich in [MM02]. Die grundsätzliche Idee ist in Abbildung Abbildung 5.5 angedeutet.

Bei der Bestimmung des Richtungsvektors besteht weiterhin Forschungsbedarf. In [KdHA04] wurde zum Beispiel ein modifizierter Ansatz des steilsten Abstiegs verwendet, der sich adaptiv an das Verhalten der Zielfunktion anpasst, wodurch in einigen Fällen der steilste Abstieg besser approximiert wird und die Schrittweite dynamisch angepasst werden kann. Ein ähnlicher Ansatz wird in [Kle06] vorgeschlagen und **Generalized RSM** genannt. Dabei wird die Methode des steilsten Abstiegs in Anlehnung an innere Punkt Methoden [FM68] modifiziert.

### 5.5.4. Phase 2

In der zweiten Phase werden zur Optimierung lineare Regressionsmodelle zweiter Ordnung (siehe Kapitel 4) eingesetzt. Diese benötigen mehr bekannte Punkte im lokalen Suchbereich, weshalb jetzt zentral zusammengesetzte Designs (siehe Kapitel 3) genutzt werden. Auf Basis der so gewählten Designpunkte wird mittels der Methode der kleinsten Fehlerquadrate ein Regressionsmodell zweiter Ordnung angepasst und darüber der stationäre Punkt $x_s$ berechnet. Je nach Optimierungsziel kann $x_s$ verworfen werden, wenn es sich nicht um ein Maximum bzw. Minimum handelt. Ansonsten muss $x_s$ ausgewertet und der Messwert mit dem Messwert des aktuell besten Punkts verglichen werden. Darüber hinaus ist zu beachten, dass der Punkt $x_s$ im aktuellen lokalen Suchbereich liegen sollte, da sonst die Approximation sehr ungenau ist.

### 5.5.5. Möglichkeiten zur Überprüfung der Güte von Regressionsmodellen

In der bisherigen Beschreibung wurde davon ausgegangen, dass die Modelle erster bzw. zweiter Ordnung eine gute Approximation der (gemessenen) Wirklichkeit darstellen. Vor allem bei starker Störung der Messwerte ist dies jedoch nicht immer der Fall. Das Ergebnis eines Optimierungsschritts auf der Basis einer schlechten Approximation ist in den meisten Fällen ein Fehlschlag. Die Bewertung der Qualität einer Approximation ermöglicht entsprechend darauf zu reagieren, wobei vier Möglichkeiten zur Verfügung stehen:

- Abbruch des Optimierungsschritts
- Erneute ggf. genauere Auswertung der Messwerte der aktuellen Designpunkte
- Auswertung zusätzlicher Designpunkte
- Approximation eines Modells höherer Ordnung

Häufig werden dazu die Methoden **Lack of Fit Tests** und **Coefficient of Determination** genutzt. Diese sind in [MM02] genauer beschrieben.

Bei beiden Verfahren ist jedoch zu beachten, dass die Auswertung eines Designs vor allem bei mehr als zwei Faktoren aufwändig ist und die Auswertung eines neuen besten Punkts einen relativ geringen Aufwand mit sich bringt. Die Tests sollten also so durchgeführt werden, dass nur besonders schlechte Modelle verworfen werden. Zu beachten ist auch, dass diese Verfahren dazu führen können, ein schlechteres Endergebnis zu erhalten, wenn ausreichend gute Modelle verworfen werden.

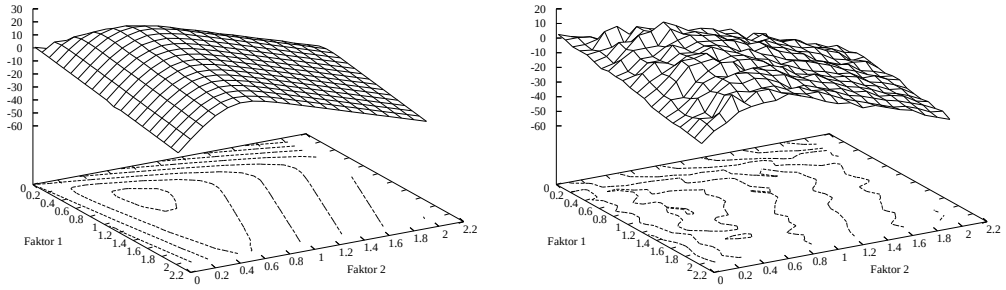



Abbildung 5.6.: Aussehen der Oberfläche bei 100 Replikationen (links) und beobachtete Oberfläche ohne Replikationen (rechts)

### 5.5.6. Anpassung an Simulationsmodelle und numerische Verfahren

In diesem Abschnitt werden zwei Anpassungen von RSM auf spezielle Situationen beschrieben. Zum einen ist dies die Verwendung von Simulationen zum anderen die Verwendung von numerischen Verfahren.

RSM kann durch eine Erhöhung der Replikationen auf Simulationsmodelle angepasst werden. Im Allgemeinen wird hier ein Verhältnis von 3:1 für Auswertungen des besten Punkts und der Designpunkte empfohlen [MM02]. Dies kann jedoch nur als „Daumenregel“ verstanden werden, die in vielen Situationen gute Ergebnisse ermöglicht. Insbesondere in Situationen mit geringen Schwankungen in den Messwerten können Verhältnisse von 2:1 oder 1:1 in kürzerer Zeit zu vergleichbaren Ergebnissen führen. In [BMT05] wurde diese Empfehlung genauer untersucht. Als Grundlage wurde ein Prozesskettenmodell nach dem ProC/B Paradigma erstellt, das dem zweidimensionalen Fall der Produktionslinie aus Kapitel 2.2.1 entspricht.

In dem Modell konnten die durchschnittlichen Bedienraten $w = (w_1, w_2)'$ der beiden Server eingestellt werden. Für einen Punkt $w$ berechnet sich der Wert der Zielfunktion $R$ wie folgt:

$$R(w) = r \cdot C(w) - c \cdot w \tag{5.5}$$

Dabei bestimmt $C$ mittels Simulation den Durchsatz an Server 2; also die von Server 2 bedienten Einheiten je Zeiteinheit. $r$ entspricht den Einnahmen je produzierter Einheit und der Vektor $c = (c_1, c_2)$ den Kosten der Server. Sowohl die Gesamtkosten $c \cdot w$ als auch die Einnahmen $r \cdot C(w)$ sind dementsprechend von $w$ abhängig. In Abbildung 5.6 ist der Verlauf der Funktion für den Wertebereich [0;2] für beide Faktoren dargestellt. Da das Modell stochastische Elemente enthält, wurde für die linke Abbildung die Messung für jeden Punkt 100 mal wiederholt und der Mittelwert ermittelt. In der rechten Abbildung ist ein Ergebnis ohne Replikationen dargestellt.

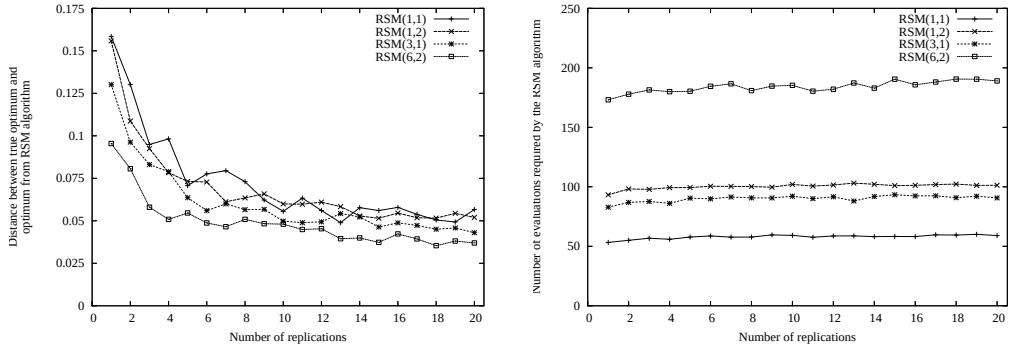



Abbildung 5.7.: Distanz des besten gefundenden Punkts zum Optimum (links) und Anzahl benötigter Replikationen (rechts)

RSM wurde bei allen Versuchen identisch konfiguriert, lediglich die Anzahl der Auswertungen am aktuell besten Punkt und an den Designpunkten war unterschiedlich. Die Konfigurationen werden mit $RSM(a; b)$ beschriftet, wobei $a$ die Anzahl der Auswertungen des besten Punkts und $b$ die Anzahl der Auswertungen der Designpunkte ist. Darüber hinaus wurde die Anzahl der Replikationen der Simulation je Auswertung durch RSM von 1 bis 20 variiert. In Abbildung 5.7 sind die Ergebnisse der Versuche im Mittel über 100 Wiederholungen von RSM dargestellt. Es ist deutlich zu sehen, dass zusätzliche Replikationen das Ergebnis verbessern, da im linken Teil der Abbildung die Distanz des besten gefundenen Punkts zum Optimum mit steigender Anzahl Replikationen fällt. Darüber hinaus wird deutlich, dass $RSM(1; 2)$ den anderen Konfigurationen unterlegen ist, da es nicht deutlich bessere Ergebnisse als $RSM(1; 1)$ und schlechtere Ergebnisse als $RSM(3; 1)$ produziert, jedoch mehr Auswertungen benötigt (siehe Abbildung 5.7 rechts).

Auf der Basis dieser Ergebnisse scheint es nicht sinnvoll die Designpunkte häufiger auszuwerten als den jeweils besten Punkt. Abhängig von der Situation scheint ein Verhältnis von 3:1 bis 1:1 sinnvoll. Da die Ergebnisse vom zu Grunde liegenden Modell abhängen können jedoch keine allgemeineren Aussagen getroffen werden.

Eine Anpassung von RSM für numerische Verfahren wurde in [KMT06] anhand eines Warteschlangennetzes vorgestellt. Zuerst wurden drei Strategien entwickelt, die die benötigte Rechenzeit reduzieren sollten, ohne die Qualität der Ergebnisse wesentlich zu verschlechtern. Bei der numerischen Analyse kann die Rechenzeit über die Anzahl der Iterationen des numerischen Lösers bis zum Erreichen der Konvergenzkriterien bestimmt werden. Im Gegensatz zum vorherigen Beispiel wird hier daher nicht die Anzahl der Auswertungen sondern die Summe der Iterationen des numerischen Lösers betrachtet.

Bei der numerischen Analyse muss eine Startverteilung $\pi_{0,x}$ gewählt werden. Häufig wird dazu eine Gleichverteilung oder eine andere fest vorgegebene Verteilung genutzt.

1: $\pi^{(0)} := \pi_0$;
2: Starte den Gauss-Seidel Löser mit $\pi^{(0)}$ und iteriere bis das maximale Residuum $d_2$ kleiner als $\varepsilon = 10^{-1}$ ist.
3: Berechne die Zielfunktion $y^{(0)} := f(w, \pi^{(0)})$.
4: $i := min\{j | 10^{-j} \leq d_2\}$;
5: **repeat**
6: $\quad i := i + 1$;
7: $\quad$ Starte den Gauss-Seidel Löser mit $\pi_0 := \pi^{(i-1)}$ bis das maximale Residuum $d_2$ kleiner als $\varepsilon = 10^{-i}$ ist.
8: $\quad$ Berechne die Zielfunktion $y^{(i)} := f(w, \pi^{(i)})$.
9: **until** $|(y^{(i-1)} - y^{(i)})/y^{(i-1)}| < 10^{-2}$;
10: $k := i$;
11: **repeat**
12: $\quad i := i + 1$;
13: $\quad$ Starte den Gauss-Seidel Löser $\pi_0 := \pi^{(i-1)}$ bis das maximale Residuum $d_2$ kleiner als $\varepsilon = 10^{-(k+(i-k)/10)}$ ist.
14: $\quad$ Berechne die Zielfunktion $y^{(i)} := f(w, \pi^{(i)})$
15: **until** $|(y^{(i-1)} - y^{(i)})/y^{(i-1)}| < 10^{-4}$;
16: **return** $\varepsilon_0 := \varepsilon$;

**Algorithmus 5.2:** Pseudocode zum Inspektionsschritt von Strategie 2

Da die Lösungsvektoren eines Modells für verschiedene Punkte in der Regel eine große Ähnlichkeit aufweisen, erscheint es sinnvoll, den Lösungsvektor $\pi_x$ eines bereits berechneten Punkts als Startverteilung für einen neu zu berechnenden Punkt zu verwenden:

**Strategie 1:** Als initiale Verteilung $\pi_{0,x}$ für einen Punkt $x$ wird der Lösungsvektor $\pi_{x'}$ des nächst gelegenen Punkts $x'$ verwendet.

Neben diesem Ansatz kann die Anzahl der Iterationen ggf. gesenkt werden, indem die benötigte Genauigkeit zu Beginn des Algorithmus bestimmt wird.

**Strategie 2:**

1. Inspektionsschritt: Bestimme die benötigte Genauigkeit $\varepsilon$ an den Eckpunkten des initialen lokalen Bereichs von RSM wie in Algorithmus 5.2 dargestellt.
2. Hauptschritt: Werte alle Punkte nur mit der Genauigkeit $\varepsilon$ aus, um Iterationen zu sparen.

Da sich die benötigte Genauigkeit ggf. im Laufe der Optimierung ändert, kann es vorkommen, dass Strategie 2 zu ungenau wird. Um dies zu verhindern, passt Strategie 3 die Genauigkeit bei Bedarf an.

**Strategie 3:** Starte mit einer initialen Genauigkeit von $\varepsilon_0$ und aktualisiere die Genauigkeit, wenn RSM den lokalen Bereich deutlich verkleinert.

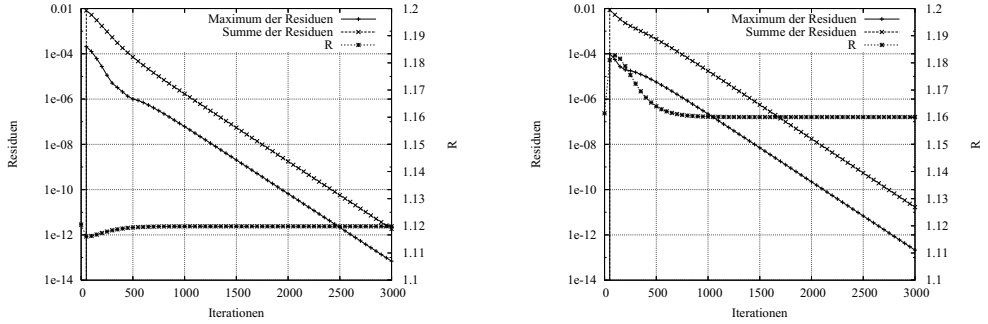



Abbildung 5.8.: Verhalten der gemessenen Werte bei wechselnder Anzahl Iterationen des numerischen Lösers an den Punkten $(0,4;0,4)$ (links) und $(0,6;0,5)$ (rechts)

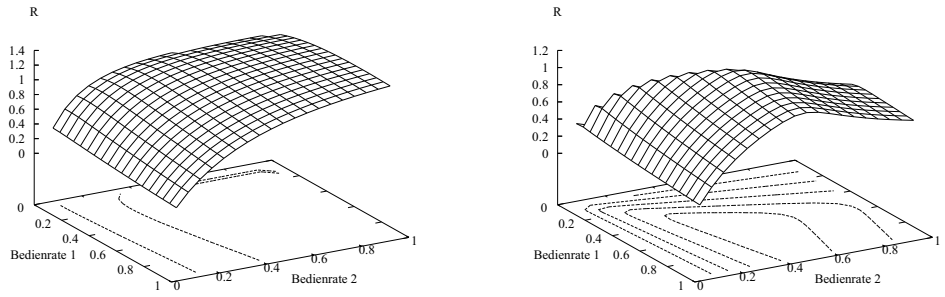



Abbildung 5.9.: Oberflächen des Modells: Berechnung mit numerischem Löser und $\varepsilon = 10^{-2}$ (links) sowie $\varepsilon = 10^{-10}$ (rechts)

Die Experimente werden analog mit zwei in Reihe angeordneten Warteschlangen durchgeführt. Die Funktion $R$ ist jedoch leicht modifiziert:

$$R(w) = \frac{r \cdot C(w)}{c_0 + c \cdot w} \tag{5.6}$$

Zu den bekannten Werten Einnahmen je Stück $r$ und Kostenvektor $c$ sind hier noch die Fixkosten $c_0$ hinzugekommen.

Vor der eigentlichen Bewertung der Strategien bzgl. dem Verhalten von RSM werden noch einige zusätzliche Untersuchungen durchgeführt. In Abbildung 5.8 wird die Entwicklung von $R$ und den Residuen in Abhängigkeit von der Anzahl Iterationen des numerischen Lösers für die Punkte $(0,4;0,4)$ und $0,6;0,5$ dargestellt. Es ist

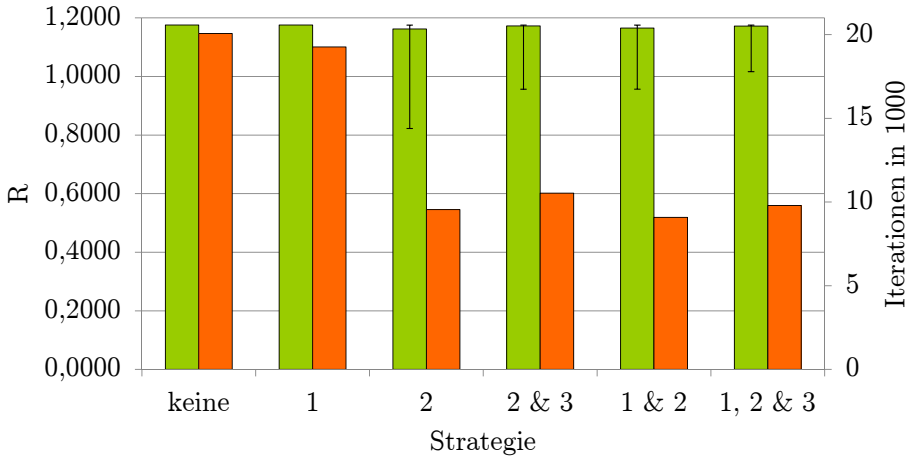



Abbildung 5.10.: Ergebnisse von RSM mit numerischem Löser und verschiedenen Strategien für die Zielfunktion $R$

deutlich zu sehen, dass $R$ bereits bei sehr hohen Residuen konvergiert. Dies würde bei den Strategien 2 und 3 zu deutlichen Einsparungen führen.

In Abbildung 5.9 wird der Verlauf der Funktion $R$ abhängig von der Genauigkeit $\varepsilon$ dargestellt, die in der linken Darstellung $10^{-2}$ und in der rechten Darstellung $10^{-10}$ beträgt. Es ist offensichtlich, dass sich der Verlauf der Funktion signifikant unterscheidet. Daher müssen die Strategien eine ausreichende Mindestgenauigkeit bestimmen, um ein passende Bestimmung der Funktionswerte zu gewährleisten.

In Abbildung 5.10 sind die Ergebnisse der Optimierung mit RSM über 100 Replikationen aufgeführt. Die Balken beziehen sich dabei auf die Mittel der gemessenen Werte, wobei sich die grünen Balken auf die linke Skala und die orangen Balken auf die rechte Skala beziehen. Die grünen Balken geben den durchschnittlichen Funktionswert des besten gefundenen Punkts an, zusätzlich sind dort die minimalen und maximalen Werte verzeichnet. Die orangen Werte kennzeichnen die Anzahl der benötigten Iterationen des numerischen Lösers. Die sechs Balkengruppen entsprechen dabei den unterschiedlichen Strategien:

- keine: RSM ohne Einsatz der Strategien
- 1: RSM mit Strategie 1
- 2: RSM mit Strategie 2
- 2 & 3: RSM mit den Strategien 2 und 3
- 1 & 2: RSM mit den Strategien 1 und 2
- 1,2 & 3: RSM mit den Strategien 1, 2 und 3

Strategie 1 reduziert die Anzahl der Iterationen im Vergleich zum normalen RSM bei gleich bleibender Ergebnisqualität leicht. Strategien 2 und 3 können die Anzahl der Iterationen deutlich reduzieren, wodurch jedoch die Ergebnisqualität leicht sinkt. Deutlich wird hier auch, dass die erneute Anpassung der Genauigkeit bei Strategie 3 zu einer leicht höheren Anzahl Iteration im Vergleich zu Strategie 2 führt, gleichzeitig aber auch bessere Ergebnisse liefert. Die Kombination von Strategie 1 mit den Strategien 2 und 3 führt zu einer leichten Reduktion der Iterationen. Überraschend ist jedoch, dass darüber hinaus Differenz zwischen dem maximalen und minimalen gemessenen Werten reduziert wird. Insgesamt zeigt sich damit, dass dieser Ansatz für die Optimierung mit numerischer Analyse interessant ist und Verbesserungen bietet.

## 5.6. Problemfälle der lokalen Verfahren

In diesem Abschnitt werden zwei Situationen vorgestellt, in denen jeweils eines der vorgestellten lokalen Verfahren dem Anderen deutlich unterlegen ist. Durch diese Gegenüberstellung soll verdeutlicht werden, in welchen Bereichen die einzelnen Verfahren Vor- und Nachteile aufweisen.

Zuerst wird eine Situation betrachtet, in der RSM im Allgemeinen effizienter als PS ist. Gegeben sei dazu die Funktion $f_1$ mit $x_i \in [-1, \ldots, 1]$ mit $i = 1, 2$, die in Abbildung 5.11 dargestellt ist:

$$f_1(x) = (x_1 - x_2)^2 + 0,01 \cdot (x_1 + x_2)^2 \tag{5.7}$$

Die Funktionswerte dieser Funktion steigen bei Abweichung von der durch $x_1 = x_2$ gegebenen Diagonale stark an, so dass sich sowohl RSM als auch PS in den ersten Schritten dieser Diagonalen annähern. Nun wird die Situation betrachtet, in der die Verfahren den Punkt $x = (0,5; 0,5)$ mit Funktionswert $0,01$ erreichen. Bei einer Halbbreite des lokalen Bereichs betrachtet PS die Punkte $(0,4; 0,5)$, $(0,6; 0,5)$, $(0,5; 0,4)$ und $(0,5; 0,6)$ mit den Funktionswerten $0,0181$, $0,0221$, $0,0181$ und $0,0221$. Da alle Funktionswerte größer der Funktionswert $f_1(x)$ sind, wird die Halbbreite verkleinert. In diesem Beispiel muss die Halbbreite dreimal halbiert werden, bevor einer der benachbarten Punkte einen niedrigeren Funktionswert aufweist. Die Suche nach dem Optimum wird daher in deutlich kleineren Schritten durchgeführt. Im schlimmsten Fall würde mit den Verkleinerungen ein Abbruchkriterium erfüllt und das Verfahren mit einer deutlichen Distanz zum Optimum abgebrochen. Im Gegenzug betrachtet RSM die Punkte $(0,400; 0,400)$, $(0,400; 0,600)$, $(0,600; 0,400)$ und $(0,600; 0,600)$ mit den Funktionswerten $0,0064$, $0,0500$, $0,0500$ und $0,0144$ und bestimmt die Richtung des steilsten Abstiegs als $(-1; -1)$. Mit diesem Richtungsvektor wird das Optimum bereits nach 5 weiteren betrachteten Punkten erreicht.

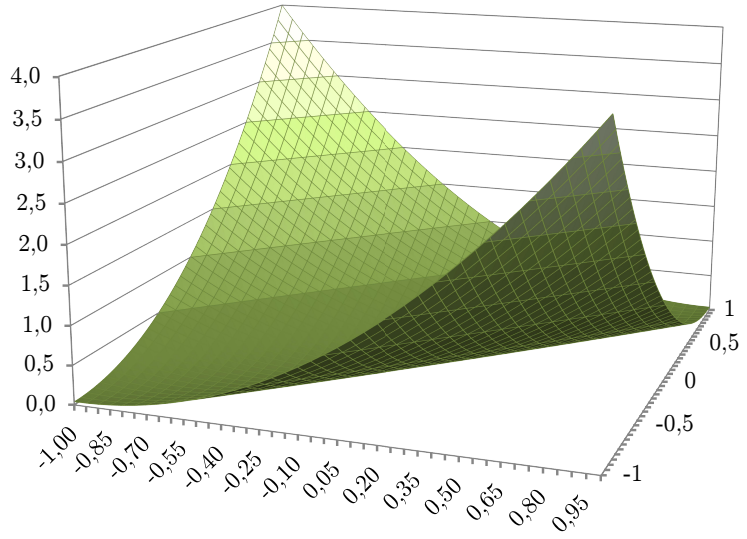



Abbildung 5.11.: Grafische Darstellung von $f_1$

Im Gegensatz zum letzten Beispiel kann PS in einer ähnlichen Situation zu besseren Ergebnissen führen. Die Funktion $f_2$ ist der Funktion $f_1$ - um 45 Grad gedreht - sehr ähnlich und in Abbildung 5.12 dargestellt:

$$f_2(x) = x_1^2 + 0,03 \cdot x_2^2 \tag{5.8}$$

Die Funktion $f_2$ könnte für die beiden Dimensionen einzeln optimiert werden und besitzt damit besonders gute Voraussetzungen, um das Optimum mittels PS zu bestimmen, wobei der Startpunkt für PS in diesem Fall unerheblich ist. Für RSM besteht hier jedoch ein Problem, wenn für die Halbbreite $w$ des Suchbereichs und den aktuell besten Punkt $x$ gilt, dass $|x_1| - 0,5 \cdot w \leq \varepsilon$ mit kleinem $\varepsilon > 0$ ist. Der Abstand für den ersten Faktor zum Wert an der Stelle des Optimums ist dementsprechend halb so groß wie $w$. Dies führt zu einer Richtung des steilsten Abstiegs von $(1 \pm \varepsilon_1; -\varepsilon_2)$ oder $(-1 \pm \varepsilon_1; -\varepsilon_2)$ mit kleinen $\varepsilon_i > 0$ mit $i = 1, 2$. Mit diesem Richtungsvektor wird bei ausreichend großem $\varepsilon_2$ ein besserer Punkt $x'$ mittels RSM gefunden, wobei $x'_1$ fast gleich $-x_1$ ist. Auf diese Weise springt der Wert des ersten Faktors von der einen Seite des optimalen Werts zur anderen, wobei die Distanz zum optimalen Wert im wesentlichen beibehalten wird. Der Wert des zweiten Faktors verändert sich hingegen nur geringfügig. Daraus folgt, dass nach einem Schritt mittels RSM, die Situation im wesentlichen analog zu der vor diesem Schritt ist, wobei sich der Funktionswert nur geringfügig verbessert hat. Daher wird mit hoher Wahrscheinlichkeit der nächste Schritt fast in die entgegengesetzte Richtung durchgeführt. Der daraus entstandene Punkt $x''$ liegt nahe dem ursprünglichen Punkt

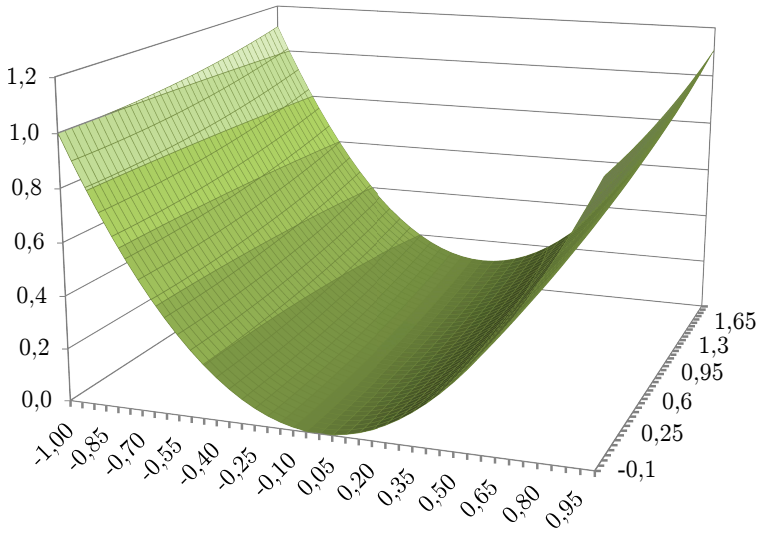



Abbildung 5.12.: Grafische Darstellung von $f_2$

$x$. Da sich dieses Verhalten entsprechend fortsetzt, werden unverhältnismäßig viele Experimentdesigns bestimmt und Punkte ausgewertet.

Eine Lösung für dieses Problems besteht in der Überprüfung, ob der aktuelle Punkt nahe dem Ergebnis des vorletzten Schritts liegt. Ist dies der Fall, wird die Halbbreite $w$ des Suchbereichs reduziert, so dass im nächsten Schritt der Wert des ersten Faktors sich stark dem optimalen Wert annähert. Da das beschriebene Problem bei verschiedenen Modellen, z.B. auch bei dem Lagerhaltungsmodell (Abschnitt 2.2.2), auftreten kann, wird diese Strategie häufig genutzt.

## 5.7. Evolutionäre Algorithmen

EA leiten sich von der Evolutionstheorie ab und werden an verschiedensten Stellen eingesetzt. Populär sind sie vor allem in Bereichen, in denen die Auswertung von Funktionen nicht kostenintensiv ist, da sie häufig deutlich mehr Punkte auswerten als andere Verfahren, auch wenn speziell angepasste Varianten existieren. Eine allgemeine Beschreibung der EA finden sich z.B. in [Sch95]. Die Beschreibung der EA ist bewusst kurz gehalten, da dieses Verfahren nicht genauer untersucht wird. EA können jedoch bei der Optimierung mit Kriging Metamodellen (siehe Kapitel 6) genutzt werden, daher wird hier die grundlegenden Vorgehensweisen beschrieben.

Der zentrale, den EA zugrunde liegende Teil der Evolutionstheorie (siehe Abbildung 5.13) besteht darin, dass es eine Menge von Individuen einer Spezies gibt.

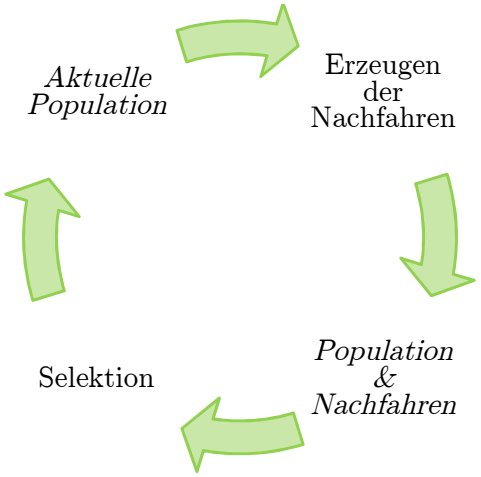



Abbildung 5.13.: Vereinfachte Darstellung der Evolution

Diese ist die aktuelle Population. Diese Population erzeugt eine Reihe von Nachfahren. Dabei unterscheiden sich die Nachfahren in der Regel nur geringfügig von ihren Vorfahren. Aufgrund des Evolutionsdrucks überleben nicht alle Individuen, so dass eine Selektion stattfindet, bei der die schlechtesten Individuen sterben. Die Überlebenden bilden dann eine neue Population und der Zyklus beginnt erneut.

Entsprechend dieser Theorie nutzen EA eine Population $P$ von $\mu$ Individuen, die $\lambda$ Nachfahren $N$ mittels verschiedener Operatoren produzieren. Aus der bestehenden Population und den Nachfahren wird eine neue Population der stärksten Individuen ermittelt. Dabei ist zu beachten, dass die Lebenszeit eines Individuums in der Regel durch eine Konstante $t$ begrenzt ist. Für $t = \infty$ sterben Individuen der aktuellen Population nicht und man spricht von einer Plus-Strategie. Ist $t = 0$, sterben Individuen der aktuellen Population auf jeden Fall und die nachfolgende Population wird ausschließlich aus den Nachfahren der aktuellen Population gebildet. In diesem Fall spricht man von einer Komma-Strategie.

Die Standardoperatoren zur Berechnung der Nachfahren sind die Mutation und die Rekombination. Für beide Operatoren existieren verschiedene Umsetzungen, die jeweils auf unterschiedliche Problemstellungen ausgerichtet sind. Die folgende Beschreibung orientiert sich daher an den Ausführungen in [AB02].

Bei der Mutation wird ein Individuum aus der aktuellen Population kopiert. Diese Kopie wird zufällig an einzelnen Stellen modifiziert. Die Mutation wird für einen Punkt $x$ durchgeführt, indem zuerst der Mutationsvektor $z$ berechnet wird. Die einzelnen Werte des Vektors sind dabei normalverteilt mit Mittelwert 0 und Varianz $\sigma$. Der Nachfolger $x'$ des Punkts $x$ berechnet sich daher wie folgt:

$$x' = x + z \tag{5.9}$$

In der Regel wird die Stärke der Mutation im Verlauf der Optimierung reduziert. Beim zweiten Operator, der Rekombination, wird ein neuer Punkt als Mittelwert von $\rho$ Individuen berechnet. Es werden dazu nacheinander $\rho$ gleichverteilte Zufallszahlen aus $\{1, \ldots, \mu\}$ gezogen und die zugehörigen Individuen für die Berechnung genutzt. Dabei kann das gleiche Individuum mehrfach in der Berechnung berücksichtigt und so stärker gewichtet werden. Wenn $a = \{a_1, \ldots, a_\rho\}$ eine Multimenge der Indizes ist und wie gerade beschrieben erzeugt wurde, berechnet sich der Nachfahre $x'$ wie folgt:

$$x' = \frac{1}{\rho} \sum_{i=1}^{\rho} P_{a_i} + z \tag{5.10}$$

Dabei ist $P_i$ das $i$-te Individuum der Population $P$. Eine Ausnahme bildet der Fall $\rho = \mu$. Hier werden alle Individuen zur Berechnung des Nachfolgers eingesetzt:

$$x' = \frac{1}{\mu} \sum_{i=1}^{\mu} P_i + z \tag{5.11}$$

Für die Auswahl einer bestimmten Anzahl $m$ von Individuen aus einer Menge der Größe $n \geq m$ existiert eine Reihe von Methoden. In [BT05] werden bekannte Methoden auf ihre Einsetzbarkeit mit EA untersucht und ein neues Verfahren vorgeschlagen. Eine genaue Beschreibung dieser Verfahren erfolgt in Kapitel 9.3.1.

EA können effizient in Situationen eingesetzt werden, bei denen sowohl schnelle approximative als auch zeitaufwändige exakte Verfahren existierten. In dieser Situation können die Individuen zuerst mit dem approximativen Verfahren analysiert und auf dieser Basis viel versprechende Individuen mit dem exakten Verfahren analysiert werden. Auf diese Weise kann die Anzahl der Auswertungen mit dem exakten Verfahren reduziert und damit die Rechenzeit verringert werden. Eine aktuelle Arbeit zu diesem Thema mit dem Fokus auf Markov-Modelle ist [BK06].

# 6. Optimierung mit Kriging Metamodellen

Bisher wurden Suchheuristiken für unimodale und multimodale deterministische bzw. stochastische Funktionen vorgestellt. Diese Verfahren haben jedoch einige Nachteile wie z.B die Beschränkung auf unimodale Funktionen oder die Notwendigkeit einer sehr hohen Anzahl an Auswertungen, die vor allem beim Einsatz ereignisdiskreter Modelle mit hohem Analyseaufwand zum Tragen kommen. Die Optimierung mit Kriging Metamodellen ist in den letzten Jahren deutlich bekannter geworden und bietet in vielen Fällen gute Optimierungergebnisse mit einer geringen Anzahl an Auswertungen. Die Benutzung von Kriging Metamodellen in der Simulation und bei der Optimierung von ereignisdiskreten Simulationsmodellen ist jedoch ein relativ neues Feld. Die folgenden Abschnitte befassen sich daher zuerst mit der Erstellung von Kriging Metamodellen und anschließend mit der Optimierung auf Basis dieser Metamodelle. Die folgenden allgemeinen Beschreibungen orientieren sich an [Cre93]. Die Notation sowie die Beschreibung der Optimierung mittels KM orientiert sich an [JSW98, Jon01]. An passenden Stellen werden interessante Alternativen bzw. mögliche Veränderungen aufgeführt.

## 6.1. Optimierung mittels Kriging Metamodellen

KM bieten eine gute Approximation einer Funktion $f$. Da die Berechnung von $\hat{y}(x)$ auf der Basis des KM in der Regel deutlich weniger Rechenzeit beansprucht als die Auswertung von $f(x)$, können mehr Punkte untersucht werden. Dadurch können Optimierungsverfahren wie zum Beispiel Evolutionäre Algorithmen genutzt werden, die im Normalfall für Simulationsmodelle aufgrund der hohen Anzahl von Auswertungen nicht eingesetzt werden.

In den folgenden Abschnitten liegt der Fokus weiterhin auf einkriterieller Optimierung. KM sind jedoch nicht darauf beschränkt, wie zum Beispiel [DM05] und [HS08] zeigen. Dabei wird z.B. für jede Zielfunktion jeweils ein KM angepasst.

Da KM zu Beginn einige ausgewertete Punkte benötigen, um eine initiale Approximation durchführen zu können, muss ein entsprechendes Experimentdesign genutzt werden, um diese Punkte zu ermitteln. Als zweckmäßig haben sich hier raumfüllende Designs wie das Latin Hypercube Sampling (siehe Kapitel 3.2.4) erwiesen. Nach der Bestimmung des initialen KM wird iterativ ein neuer Punkt $x_{next}$ auf der Grundlage

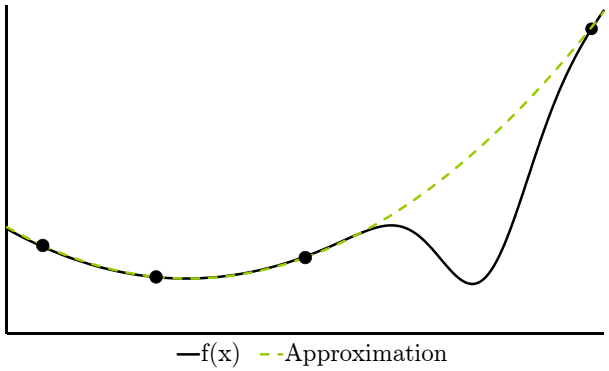



Abbildung 6.1.: Beispiel für den Verlauf einer Funktion mit vier ausgewerteten Punkten und einer Approximation durch ein Kriging Metamodell

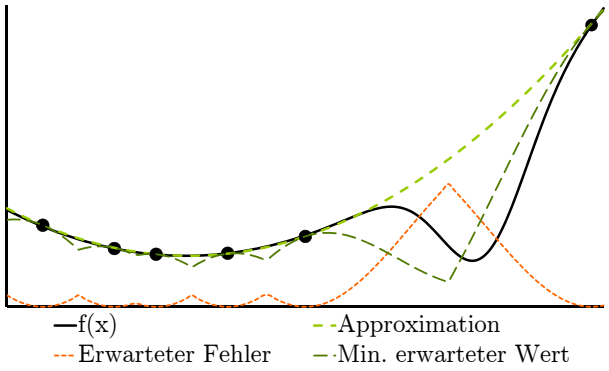



Abbildung 6.2.: Erweiterung von Abbildung 6.1 durch den maximalen erwarteten Fehler und den minimalen erwarteten Funktionswert

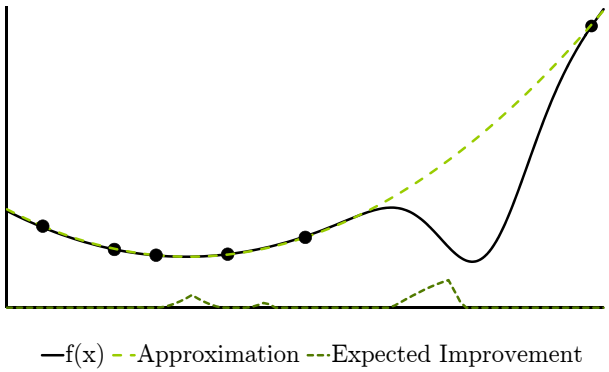



Abbildung 6.3.: Fortführung des Beispiels aus Abbildung 6.2 mit Angabe des Expected Improvements

des Metamodells ermittelt. Dieser wird ausgewertet und dem Metamodell hinzugefügt.

Ein trivialer Ansatz für die Bestimmung von $x_{next}$ wäre nun, mittels geeigneter Optimierungsverfahren das Optimum von $\hat{y}(x)$ zu bestimmen. Dieser Ansatz könnte jedoch bisher wenig untersuchte Regionen des Suchbereichs vernachlässigen. Aufgrund einer schlechten Anpassung dieser Bereiche kann es daher dazu kommen, dass Optima nicht gefunden werden (siehe Abbildung 6.1). Dieses Problem kann mit dem Schätzer für den erwarteten Fehler $s^2$ reduziert werden. Durch eine Kombination der Approximation $\hat{y}(x)$ und dem erwarteten Fehler $s^2(x)$ kann der niedrigste erwartete Wert an einer Stelle $x$ berechnet werden (siehe Abbildung 6.2).

In der Regel wird das das **Expected Improvement** (EI) verwendet, das sich aus dieser Überlegung ergibt und ein Maß für die Wahrscheinlichkeit ist, dass ein Punkt einen niedrigeren Funktionswert als den niedrigsten bisher bekannten Funktionswert besitzt (siehe Abbildung 6.3). In den folgenden Abschnitten wird das EI formal festgelegt. Zentraler Wert ist dabei der niedrigste bisher bekannte Funktionswert $f_{min}$:

$$f_{min} = min\left(y(x_1), \ldots, y(x_n)\right) \tag{6.1}$$

Entsprechend der Annahme, dass der Funktionswert von $f$ an der Stelle $x$ durch $\hat{y}(x)$ approximiert wird und der Schätzfehler ein normalverteilter Fehler mit dem Wert $s^2(x)$ ist, wird $Y_x$ als eine Zufallsvariable für die Stelle $x$ mit dem Mittelwert $\hat{y}(x)$ und Varianz $s^2(x)$ definiert. Die Verbesserung an einem Punkt $x$ berechnet sich unter Verwendung der Zufallsvariablen $Y_x$ wie folgt:

$$I(x) = max(f_{min} - Y_x, 0) \tag{6.2}$$

Daraus ergibt sich der Erwartungswert für $I(x)$ als $E[I(x)] = E[max(f_{min} - Y_x, 0)]$. Durch Umformung erhält man die Formel:

$$E[I(x)] = (f_{min} - \hat{y}(x)) \cdot \Phi\left(\frac{f_{min} - \hat{y}(x)}{s(x)}\right) + s(x) \cdot \phi\left(\frac{f_{min} - \hat{y}(x)}{s(x)}\right) \tag{6.3}$$

Dabei stehen $\Phi$ und $\phi$ für die Verteilungs- und die Dichtefunktion der Standardnormalverteilung. In Anlehnung an die Abkürzung von Expected Improvement wird für $E[I(x)]$ häufig $EI(x)$ verwendet.

Die Bestimmung des maximalen EI ist wieder eine Optimierungsaufgabe. Im Gegensatz zur ursprünglichen Optimierung ist hier die Auswertung einzelner Punkte weniger zeitintensiv und bestimmte Eigenschaften dieser Funktion bekannt sind. In [JSW98] werden diese Informationen genutzt, um einen Branch-and-Bound-Algorithmus für dieses Problem zu entwerfen. Gleichzeitig wird dort die Optimierung

eingeschränkt, indem $\rho_i = 1$ für $i = 1, \ldots, k$ fest gesetzt wird. Abhängig von der Anzahl der Dimensionen wird eine randomisierte Gitternetzsuche oder Evolutionäre Algorithmen für diese Optimierung eingesetzt. Diese Strategie wurde zum Beispiel in [HANZ05] verwendet.

### 6.1.1. Erweiterungen und Alternativen des Expected Improvements

Das normale $EI$ sucht einen Mittelweg zwischen der Reduktion der Unsicherheit im Metamodell und dem Finden besserer Punkte. Dieser Mittelweg entspricht jedoch nicht immer dem gewünschten Verhalten. Aus diesem Grund wurden verschieden Alternativen zum EI und Varianten des EI entwickelt, die entweder eine bessere Steuerung seines Verhaltens erlauben oder auf andere Ziele ausgerichtet sind.

In [HS07] werden Ersatzfunktion für das EI beschrieben, mit deren Hilfe ein stärkerer Einfluss auf den Ablauf der Suche möglich ist. Dazu werden die Gewichte $g$ bzw. $w$ eingeführt. Im ersten Ansatz, der „Generalized EI Function“ (GEIF), wird mit der Konstanten $g$ gesteuert, wie stark der erwartete Fehler in die Funktion einfließt:

$$GEIF_g(x) = s^g(x) \sum_{i=0}^{g} \left( (-1)^i \binom{n}{i} u^{g-i} T_i \right)$$

$$\text{mit} \tag{6.4}$$

$$u = \frac{f_{min} - \hat{y}(x)}{s(x)},\ T_i = -\phi(u)u^{i-1} + (i-1)T_{i-2},\ T_0 = \Phi(u),\ T_1 = -\phi(u)$$

Dabei ist zu beachten, dass im Fall $g = 1$ $GEIF_1(x) = EI(x)$ gilt.

Die zweite vorgeschlagene Funktion **Weighted EI Function** (WEIF) kann sowohl dem Fehler als auch der erwarteten Verbesserung mehr Gewicht verleihen. Dies wird mit einer Konstanten $w \in [0; 1]$ gesteuert:

$$WEIF_w(x) = w\,(f_{min} - \hat{y}(x)) \cdot \Phi\left(\frac{f_{min} - \hat{y}(x)}{s(x)}\right) + (1-w)s(x) \cdot \phi\left(\frac{f_{min} - \hat{y}(x)}{s(x)}\right) \tag{6.5}$$

Auch diese Funktion ist nur eine Verallgemeinerung des normalen EI und für $w = 0,5$ gilt $WEIF_{0,5}(x) = 0,5 \cdot EI(x)$. Da die Konstante $0,5$ bei der Suche nach dem maximalen EI vernachlässigt werden kann, entspricht die Funktion für $w = 0,5$ dem normalen EI.

Zum Abschluss wird noch eine Alternative zum EI erwähnt. In [VVW07] wird eine Strategie vorgeschlagen, die sich **Informational Approach to Global Optimization** (IAGO) nennt. Diese Strategie versucht die bedingte Entropie des Modells an den Stellen möglicher Minima zu reduzieren.

```
1: Erstelle ein initiales raumfüllendes Design.
2: Werte alle Punkte des Designs aus.
3: Passe ein KM an die bekannten Punkte an.
4: if KM nicht valide then
5:    Füge einen neuen Punkt hinzu.
6:    Springe zu 2.
7: end if
8: Suche nicht bereits ausgewerteten Punkt x mit bestem Wert für ŷ(x).
9: Füge Punkt zum Metamodell hinzu.
10: if Anzahl erfolgloser Suchen kleiner als die maximal erlaubte Anzahl erfolgloser
    Suchen then
11:    Springe zu 2.
12: end if
```

**Algorithmus 6.1:** Pseudocode für das Verfahren aus [KvBvN08]

### 6.1.2. Alternative Ansätze bei der Optimierung mit Kriging Metamodellen

Die in [JSW98] beschriebene und in dieser Arbeit verwendete Methode ist die populärste im Bereich der Optimierung mit KM. In den folgenden Abschnitten werden einige alternative Herangehensweisen beschrieben, die die vielfältigen Möglichkeiten dieses Bereichs verdeutlichen.

In [KvBvN08] wird ein Verfahren vorgestellt, dass gänzlich auf die Verwendung des EI verzichtet. Die Aufgaben des EI - Identifikation von Punkten mit hohem erwarteten Fehler und Identifikation von Punkten mit besserem Funktionswert - wird hier auf zwei Teile innerhalb des Algorithmus aufgeteilt. Der erste Aspekt des normalen EI wird über die Qualität des Metamodells berücksichtigt. Die Qualität des Metamodells wird durch zusätzliche Tests überprüft. Sollte die Qualität nicht ausreichend sein, werden neue Punkte dem Metamodell hinzugefügt. Der zweite Aspekt des EI wird durch eine direkte Suche nach nicht bekannten Punkten $x$ mit besserem $\hat{y}(x)$ berücksichtigt. Der Pseudocode für das präsentierte Verfahren ist in Algorithmus 6.1 dargestellt.

Andere Ansätze betrachten die Optimierungsaufgaben innerhalb der Optimierung mit KM genauer und integrieren hier bekannte Verfahren wie z.B. Simulated Annealing [GCR08]. Teilweise wird die Integration aber auch stärker in den Mittelpunkt gestellt. In [Rat99] werden KM mit EA kombiniert. Dabei werden mittels EA auf einem KM $n$ neue Punkte ermittelt. Anschließend werden diese $n$ Punkte mit der echten Funktion $f$ ausgewertet. Die Autoren präsentieren verschiedene Strategien, die so erhaltenen Punkte zu verwenden:

1. Die $n$ Punkte bilden ein neues Kriging Metamodell.

1: Erstelle initiales KM.
2: Bestimme Minima bzw. Maxima mittels EA auf dem KM.
3: Füge die neuen Punkte dem KM hinzu und passe das KM entsprechend an.
4: Solange kein Abbruchkriterium erfüllt ist, springe zu Schritt 2.

**Algorithmus 6.2:** Pseudocode für das Verfahren aus [KIL06]

2. Die $m$ schlechtesten Punkte des Metamodells werden durch die $m$ besten Punkte der Population ersetzt.
3. $m$ zufällige Punkte des Metamodells werden durch die $m$ besten Punkte der Population ersetzt.
4. Zum aktuellen Metamodell werden die $m$ besten Punkte der Population hinzugefügt.
5. Die Populationsgröße ist $m$. Die $m$ schlechtesten Punkte des Metamodells werden durch die Population ersetzt.
6. Auswahl der $m$ besten Punkte und erneute Auswertung dieser Punkte.

$m \leq n$ ist dabei eine vor Beginn des Algorithmus festzulegende Konstante. Der Einfluss der Überlegungen aus dem Bereich der EA ist bei diesen Ansätzen nicht zu übersehen, da die Ideen dieser Ansätze den Komma- bzw. Plus-Strategien der EA entsprechen.

In [KIL06] wird das EI durch eine Fitnessfunktion ersetzt, die innerhalb eines EA zur Berechnung neuer Punkte benutzt wird. Das Verfahren ist als Pseudocode in Algorithmus 6.2 skizziert. Die eingesetzt Fitnessfunktion $y_{fit}$ berücksichtigt dabei die Grundgedanken der EA, sowohl den besten Funktionswert der Zielfunktion $y$ zu berücksichtigen als auch eine gewisse Diversität in der Population zu gewährleisten. Die Gewichtung dieser beiden Aspekte kann dabei über $w \in [0; 1]$ gesteuert werden:

$$y_{fit}(x) = w \cdot \hat{y}(x) + (1 - w) \cdot min\left(distance\left(x, x_1\right), \ldots, distance\left(x, x_n\right)\right) \tag{6.6}$$

In [SSV+04] werden KM mit der Methode der **Divided Rectangles** (DIRECT) verknüpft. Bei diesem Verfahren wird ein aktueller Suchbereich (zu Beginn ist dies der gesamte Suchbereich) in zwei Hälften (Rechtecke) geteilt. Die entstandenen Hälften sind mögliche neue Suchbereiche. Welche weiter unterteilt werden, wird mittels der Lipschitz Konstante $L$ entschieden:

$$\left|f(x_1) - f(x_2)\right| \leq L \left|x_1 - x_2\right| \tag{6.7}$$

Im Verlauf des Algorithmus werden kleinere Werte von $L$ gewählt. Die führt zu einem Wechsel von globaler Suche hin zur genaueren Untersuchung bereits identifizierter Bereiche lokaler Optima.

Zwei andere Verfahren konzentrieren sich auf die Reduktion der Unsicherheit in den Modellparametern. In [Kit86] wird ein **bayesisches Kriging** vorgestellt, dass zur Untersuchung der Modellparameter bayesische Ansätze nutzt. Dieser Ansatz führt jedoch zu einem hohen Rechenaufwand, da eine Reihe numerischer Integrationen notwendig sind. Durch ein anderes Verfahren soll ein Mittelweg zwischen Rechenaufwand und Unsicherheit gewählt werden. In [Tod01] wird dafür eine modifizierte Maximum Likelihood Funktion vorgestellt.

Eine Alternative zur Bestimmung der Varianz in KM wird in [dHKS06] vorgeschlagen. Hier wird Bootstrapping eingesetzt, um die Unsicherheit des Metamodells besser zu bestimmen. Da in diesem Verfahren jedoch KM wiederholt angepasst werden müssen, steigt der Rechenaufwand stark an.

Darüber hinaus existieren Verfahren, die sich auf bestimmte Situationen beschränken. In der Dissertation von Michael Godziert [God07] wurde zum Beispiel ein globales Optimierungsverfahren unter dem Einsatz von KM entwickelt, dass vor allem dann sinnvoll eingesetzt werden kann, wenn die Auswertungen der Funktion hochgradig parallel durchgeführt werden können. In [Sas02] wird von Sasena eine Variante des Verfahrens aus [JSW98] vorgestellt, die für bestimmte Arten von Nebenbedingungen entwickelt wurde.

### 6.1.3. Untersuchung der Laufzeit

Der Rechenaufwand der Kriging Metamodelle wird vor allem durch die Invertierung der Matrix $R$, die jeweils einmal für jedes neu angepasste Metamodell durchgeführt werden muss, und durch die Matrix-Vektor-Multiplikationen mit $R$ bzw. $R^{-1}$ bestimmt, die u.a. bei der Berechnung von $\hat{y}(x)$ und $\hat{s}^2(x)$ durchgeführt werden.

Zur Invertierung einer $n \times n$-Matrix existieren verschiedene Algorithmen. Die dieser Arbeit zu grunde liegende Implementierung des Kriging-Verfahrens nutzt die Gauss-Elimination mit einer Laufzeit von $O(n^3)$. Eine Übersicht weiterer Algorithmen sowie weiterführende theoretische Betrachtungen finden sich in [Bar05]. Demnach ist die Laufzeit der Invertierung einer $n \times n$-Matrix durch eine Funktion $w(n)$ mit $n^2 \leq w(n) < n^3$ abgeschätzt werden. Gleiches gilt für die Abschätzung der Laufzeit von Matrixmultiplikationen. Dabei ist zu beachten, dass $n$ zu Beginn des Algorithmus i.d.R. unter 10 liegt, im Verlauf des Algorithmus jedoch deutlich ansteigt. Dieser Zeitaufwand ist vor allem in Verbindung mit der Anpassung des Kriging Metamodells zu beachten, da für jede Berechnung der Likelihood-Funktion diese Invertierung durchgeführt werden muss.

In den meisten Fällen werden zur Anpassung eines Kriging Metamodells Branch-and-Bound-Verfahren [JSW98], gitternetzbasierte Verfahren oder randomisierte Algorithmen wie EA benutzt. Dabei ist man nicht auf diese Verfahren beschränkt. Es können grundsätzlich fast beliebige Suchheuristiken verwendet werden. Darüber hinaus ist zu beachten, dass das in [JSW98] vorgeschlagene Verfahren die Wahl der

| Anzahl Punkte im KM | Modellanpassung | | 10000 Auswertungen | |
|---|---|---|---|---|
| | Mittelwert | $\pm\frac{1}{2}$ Konf. | Mittelwert | $\pm\frac{1}{2}$ Konf. |
| 4 | 2,75 | 0,15 | 115,73 | 0,34 |
| 8 | 5,97 | 0,07 | 147,87 | 0,47 |
| 16 | 25,12 | 0,16 | 209,27 | 1,25 |
| 32 | 100,04 | 0,46 | 371,71 | 0,46 |
| 64 | 412,64 | 0,21 | 654,38 | 1,70 |
| 128 | 2422,41 | 0,68 | 1208,85 | 1,97 |
| 256 | 16525,16 | 16,45 | 2349,10 | 3,58 |
| 512 | 125975,07 | 1491,89 | 4631,61 | 18,11 |

Tabelle 6.1.: Benötigte Zeit in Sekunden bei Kriging Metamodellen in Abhängigkeit von der Anzahl der Punkte im Metamodell (Durchschnitt und Halbbreite des Konfidenzintervalls)

Parameter des Kriging Metamodells einschränkt, um Grenzen genauer berechnen zu können und so i.d.R. eine exponentielle Laufzeit des eingesetzten Branch-And-Bound-Algorithmus zu vermeiden.

Für alle Verfahren gilt jedoch, dass bei jeder Berechnung der Likelihood-Funktion, die Variablen des Kriging Metamodells berechnet werden müssen, wobei in jedem Fall die Matrix $R$ invertiert werden muss. Daher ergibt sich die Laufzeit für die Anpassung eines Kriging Metamodells als $O(a \cdot w(n))$, wobei $a$ die Anzahl der Auswertungen der Maximum-Likelihood-Funktion darstellt.

Mit $n_{start}$ als Anzahl Punkte zu Beginn des Algorithmus und $n_{ende}$ als Anzahl der Punkte am Ende des Algorithmus, lässt sich die gesamte Laufzeit für die Anpassung der Metamodelle wie folgt abschätzen:

$$O\left(a \cdot \sum_{i=n_{start}}^{n_{ende}} w\left(i\right)\right) \tag{6.8}$$

Ein weiterer nicht zu vernachlässigender Aspekt bei der Betrachtung der Laufzeit ist die Berechnung des Expected Improvement $EI(x)$. Diese basiert auf den Ergebnissen der Funktionen $\hat{y}(x)$ und $s^2(x)$ (siehe Kapitel 4.3.1). Die Berechnung von $\hat{y}(x)$ hat nach Formel 4.10 eine Laufzeit von $O(n)$ bei einer einmaligen Vorbereitungszeit für $c$ von $O(n^2)$. Die Berechnung von $s^2(x)$ hat nach Formel 4.11 eine Laufzeit von $O(n^2 + n)$ bei einer Vorbereitungszeit für die Berechnung von $\mathbf{1}_n' \boldsymbol{R}^{-1} \mathbf{1}_n$ von $O(n^2)$. Da die Vorbereitung nur einmal bei der Anpassung des Modells durchgeführt werden muss und kleiner als die für die Anpassung des Modells benötigte Zeit ist, wird diese hier nicht weiter betrachtet. Daher ergibt sich die Laufzeit zur Berechnung des EI wie folgt:

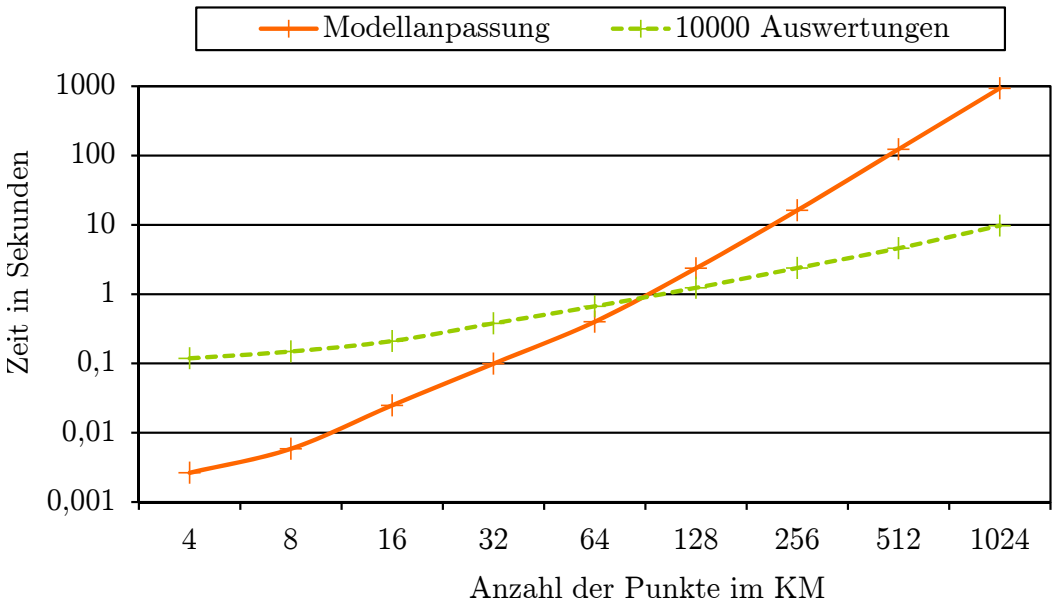



Abbildung 6.4.: Entwicklung der Zeitbedarfs von Kriging Metamodellen in Abhängigkeit der im Modell enthaltenen Punkte

$$O\left(n^2 + n\right) \tag{6.9}$$

Da das EI in jedem Schritt des Algorithmus zur Berechnung des nächsten zu untersuchenden Punkts genutzt wird und an dieser Stelle analog zur Anpassung des Kriging Metamodells eine Suchheuristik genutzt wird, ergibt sich die gesamte Laufzeit zur Berechnung des EI wie folgt, wobei $b$ die Anzahl der Auswertungen der Suchheuristik zur Bestimmung des nächsten Punkts ist:

$$O\left(b \cdot \sum_{i=n_{start}}^{n_{ende}} \left(i^2 + i\right)\right) \tag{6.10}$$

Unter Vernachlässigung weiterer Berechnungen kann die Laufzeit des gesamten Kriging-Verfahrens daher wie folgt abgeschätzt werden:

$$O\left(\sum_{i=n_{start}}^{n_{ende}} \left(a \cdot w\left(i\right) + b \cdot \left(i^2 + i\right)\right)\right) \tag{6.11}$$

Zur Veranschaulichung werden KM mit unterschiedlicher Anzahl Punkte angepasst und Auswertungen des KM durchgeführt. Die Versuche werden jeweils 100 mal durchgeführt und die entsprechenden 95% Konfidenzintervalle berechnet. Die Ergebnisse sind in Tabelle 6.1 aufgeführt und in Abbildung 6.4 visualisiert. Die Laufzeit der Auswertungen steigt dabei im Wesentlichen linear an, was auf die Berechnung nach Formel 4.10 zurückzuführen ist. Hierbei muss die Konstante $c$ nur einmal bei der ersten Auswertung berechnet werden. Durch die Anzahl von 10000 Auswertungen ist der Einfluss der Berechnungszeit für $c$ minimal, aber immer noch als quadratische Komponente vorhanden. Bei der Anpassung des KM mittels Maximum-Likelihood-Funktionen steigt der Zeitbedarf wie berechnet kubisch an.

Empirische Untersuchungen haben die Aussage der Formel 6.11 unterstützt, dass bei Werten von $n > 100$ der Rechenbedarf deutlich ansteigt. Daher ist es sinnvoll, den Rechenaufwand durch geeignete Verfahren zu reduzieren.

## 6.2. Problemstellungen bei der Optimierung mit Kriging Metamodellen

Bei der Optimierung von Funktionen mittels Kriging Metamodellen treten eine Reihe von Problemstellungen auf, die hier etwas genauer betrachten werden. Dabei wird nicht versucht, alle möglichen Probleme vollständig aufzuführen, sondern die Untersuchung auf die zentralen Aspekte beschränkt.

### 6.2.1. Initiale Experimentdesigns

Beim Einsatz von Kriging Metamodellen muss zu Beginn des Verfahrens ein initiales Experimentdesign erstellt werden. Hierzu eignen sich vor allem sogenannte raumfüllende Designs, wobei in der Regel ein sogenanntes Latin Hypercube Sampling genutzt wird. Über die Größe des initialen Experimentdesigns werden jedoch im Allgemeinen keine Aussagen gemacht. In [vBK08] werden Kriging Metamodelle zur Erstellung von Experimentdesigns eingesetzt, wobei ein minimales initiales Design genutzt wird. Auf dieser Basis werden bei den hier eingesetzten Verfahren kleine initiale Latin Hypercube Samplings als initiale Designs verwendet.

### 6.2.2. Rechenfehler

Die in Computern eingesetzte Fließkommaarithmetik unterliegt gegenüber der mathematisch exakten Berechnung einer Reihe von Einschränkungen. Dies ist vor allem darin begründet, dass die Genauigkeit der Berechnung begrenzt ist. Dies führt insbesondere beim Einsatz stark unterschiedlicher Zahlenbereiche zu Problemen.

**Beispiel:** Gegeben seien zwei Zahlen $a$ und $b$, wobei $ln|a|$ deutlich größer als $ln|b|$ ist. Je nach eingesetzter Genauigkeit kann das Ergebnis von $a + b$ gleich $a$ sein, da der Wert $a + b$ nicht darstellbar ist.

Dieses Problem kann bei Kriging Metamodellen besonders bei Berechnungen auf Basis der Korrelationsfunktion auftreten. Die Korrelation zweier Punkte wird ausgehend von der Distanz $d$ dieser beiden Punkte als $e^{-d}$ berechnet. Da die so berechnete Korrelationsmatrix $R$ und ihr Inverses $R^{-1}$ die Grundlage für viele weitere Berechnungen bilden, ist es sinnvoll, die Punkte ggf. so in **lokalen Kriging Metamodellen** zu gruppieren, dass der Unterschied der Distanzen zwischen den Punkten ähnlich ist, um Ungenauigkeiten in der Berechnung entgegen zu wirken.

### 6.2.3. Identifikation wichtiger Parameter vor der Optimierung

Dieser Bereich soll in dieser Arbeit nicht genauer betrachtet werden, trotzdem wird darauf hingewiesen, dass die hochdimensionale Optimierung ein schwierige Aufgabe ist. Können vor der Optimierung bereits Parameter identifiziert werden, die keinen oder nur einen geringen Einfluss auf die Zielfunktion haben, so kann die Entfernung dieser Parameter aus dem Optimerungsproblem die Optimierung entscheidend vereinfachen. Als Beispiel für ein solches Verfahren zur Identifikation der wichtigsten Parameter sei die **Sequential Bifurcation** genannt [KBP06].

### 6.2.4. Abbruchkriterien

Ebenfalls nicht Thema dieser Arbeit ist die Betrachtung unterschiedlicher Abbruchkriterien. Bei den späteren Betrachtungen stehen die Verfahren und deren Vergleich im Mittelpunkt. Für die Optimierung im realen Einsatz kann es durchaus von Interesse sein, die Optimierung bei Erreichen einzelner Optimalitätskriterien zu beenden, der Vergleich der Verfahren wird durch solche Abbruchkriterien jedoch erschwert. Daher beschränkt sich diese Arbeit bei den Abbruchkriterien auf die Anzahl der betrachteten Punkte, die Anzahl der im KM verwendeten Punkte sowie die Anzahl der Auswertungen. Verschiedene Optimalitätskriterien als Abbruchbedingungen in der Optimierung mit Simulationsmodellen sind in [BdCK05] beschrieben.

# 7. Hierarchische Kriging Metamodelle

In Kapitel 6.1.3 wurde die Laufzeit der KM und der KMO genauer untersucht. Dabei hat sich gezeigt, dass die benötigte Rechenzeit mit der Anzahl der Punkte im KM deutlich ansteigt. Empirische Untersuchungen haben darüber hinaus gezeigt, dass es bei einer großen Anzahl Punkte im KM bei verschiedenen Berechnungen zu deutlichen Rechenfehlern kommen kann, die durch die beschränkte Rechengenauigkeit in Computern hervorgerufen wird. Dieses Problem tritt vor allem bei der Invertierung der Matrix $R$ auf, bei der häufig Werte in stark unterschiedlichen Größenordnungen vorkommen. In diesem Kapitel werden daher verschiedene Verfahren diskutiert, die die aufgeführten Probleme vermeiden. Dabei wird ebenfalls auf die mit diesen Ansätzen einhergehenden neuen Probleme eingegangen. Im Anschluss werden die vorgestellten Ansätze empirisch untersucht.

## 7.1. Reduzierte Kriging Metamodelle

Kriging Metamodelle ermöglichen bereits mit wenigen Punkten eine gute Approximation vieler Funktionen. Im Laufe einer Optimierung wird eine Reihe von Punkten hinzugefügt, die zum Teil sehr nah beieinander liegen. Dies führt auf der einen Seite zu Problemen bei der Rechengenauigkeit, andererseits zu unnötig stark ansteigender Rechenzeit, da nah beieinander liegende Punkte in der Regel nur wenig neue Informationen über das Gesamtverhalten der Funktion liefern. Aus diesem Grund werden bei reduzierten Kriging Metamodellen nur für das Verhalten des Modells wichtige Punkte berücksichtigt.

Ein reduziertes Kriging Metamodell enthält - wenn möglich - nur dominante Punkte, wobei ein Punkt dominant heißt, wenn sein Funktionswert entweder größer oder kleiner als die Funktionswerte aller bekannten in einem Radius $r$ um diesen Punkt liegenden Punkte ist. Die Wahl dieses Kriteriums ergibt sich aus der Überlegung, dass eine Funktion vor allem durch ihre Minima und Maxima charakterisiert wird. Auch wenn diese Punkte nicht zwingend Extrema darstellen, so markieren sie in der Regel den Bereich eines Extremums. Formal dominiert daher ein Punkt $p$ alle anderen Punkte $x_i$ innerhalb des Radius $r$, wenn gilt:

$$f(p) \leq min\left\{f(x_i)\middle| \, |p - x_i| \leq r\right\} \vee f(p) \geq max\left\{f(x_i)\middle| \, |p - x_i| \leq r\right\} \tag{7.1}$$

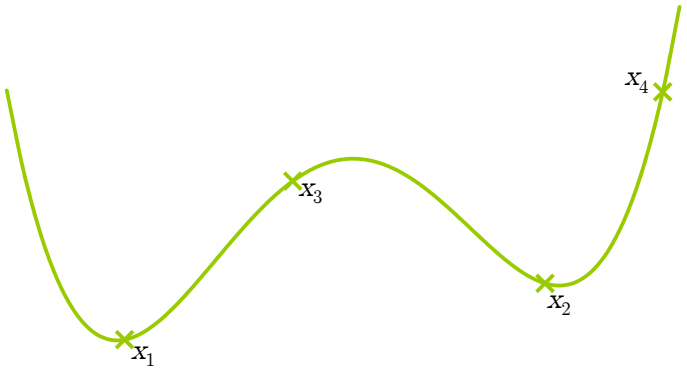



Abbildung 7.1.: Beispiel zur Abschirmungsheuristik zur Identifikation lokaler Optima: Entsprechend der Abschirmheuristik wird der Punkt $x_2$ vor dem Punkt $x_1$ durch den Punkt $x_3$ abgeschirmt. Der Punkt $x_3$ wird gleichzeitig vor dem Punkt $x_4$ durch den Punkt $x_2$ abgeschirmt.

wobei $|v|$ die euklidische Norm eines Vektors $v$ ist.

Dieses Verfahren ist darauf ausgelegt, auch bei einer hohen Anzahl Punkten eine schnelle Anpassung und Auswertung des Metamodells zu ermöglichen. Kann jedoch nur eine geringe Anzahl an dominanten Punkten identifiziert werden, führt dies zu einer schlechten Anpassung des Metamodells. Daher ist es sinnvoll, mindestens eine bestimmte Anzahl $n_{min}$ an Punkten in das Metamodell aufzunehmen, die gewährleistet, dass der Fehler in den Schätzungen gering bleibt. Zu diesem Zweck werden schrittweise Punkte ausgewählt und dem Metamodell hinzugefügt, deren minimaler Abstand zu einem der bereits gewählten Punkte maximal ist.

Bei diesem Verfahren können vor allem zwei Probleme ausgemacht werden.

**Problem 1:** Durch das Entfernen eines Punkts aus dem Kriging Metamodell besteht das Risiko, dass dieser oder ein sehr naher Punkt zur Auswertung gewählt wird. Um unnötige Auswertungen der Funktion $f$ zu vermeiden, muss in einer solchen Situation ein bereits ausgewählter aber nicht dominanter Punkt dem Metamodell hinzugefügt werden. Daraus ergibt sich zwar eine erneute Anpassung des Metamodells, diese ist i.d.R. jedoch deutlich weniger zeitintensiv als die Auswertung der Funktion $f$.

**Problem 2:** Die Wahl des Radius $r$ hat deutlichen Einfluss auf das Ergebnis des Algorithmus und ist von der zu untersuchenden Funktion abhängig. Je größer $r$ gewählt wird, desto höher ist die Wahrscheinlichkeit, nah beieinander liegende Extrema nicht ausreichend zu berücksichtigen. Je kleiner $r$ gewählt wird, desto höher ist jedoch die Wahrscheinlichkeit, dass Punkte gewählt werden, die kein Extremum darstellen, da sich z.B. keine weiteren Punkte innerhalb des Radius befinden. Aufgrund dieser Überlegungen ist es sinnvoll, sowohl typischerweise gute Werte für $r$ zu ermitteln als auch Heuristiken zu entwickeln, die diese Problematik berücksichtigen.

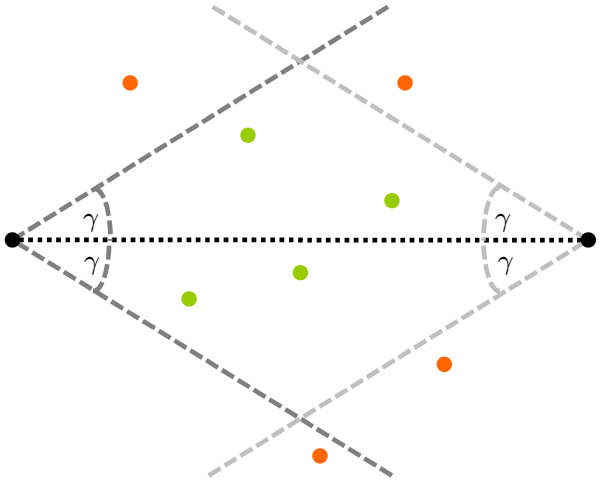



Abbildung 7.2.: Zwei Bezugspunkte (schwarz) mit Verbindungsstrecke und erlaubter Abweichung $\gamma$ sowie bzgl. Definition 17 zwischen den Bezugspunkten liegenden Punkten (grün) und außerhalb liegenden Punkten (orange)

**Problem 3:** Bei mehreren nah beieinander liegenden Extrema wird nur das ausgeprägteste berücksichtigt. Um dieses Problem zu beheben wird eine **Abschirmungsheuristik** vorgeschlagen. Zu diesem Zweck werden Punkte untersucht, die von mindestens zwei anderen Punkten dominiert werden. Dabei wird überprüft, ob diese durch mindestens einen weiteren Punkt von den dominierenden Punkten abgeschirmt werden. Anhand eines Beispiels, wie es in Abbildung 7.1 gezeigt wird, lässt sich die Situation wie folgt darstellen: Für die Punkte $x_1, x_2, x_3, x_4$ gilt $f(x_1) < f(x_2) < f(x_3) < f(x_4)$. Die Punkte $x_1$ und $x_4$ sind bei ausreichend großem $r$ in dieser Situation dominierende Punkte und die Punkte $x_2$ und $x_3$ werden von diesen Punkten dominiert. Die lokalen Extrema in der Nähe von $x_2$ und $x_3$ würden dadurch ohne diese Heuristik nicht berücksichtigt. Liegt der Punkt $x_3$ zwischen $x_1$ und $x_2$, so wird $x_2$ durch $x_3$ vor $x_1$ abgeschirmt und kann so als Punkt im reduzierten Kriging Metamodell berücksichtigt werden. Im Gegenzug wird auch $x_3$ durch $x_2$ vor $x_4$ abgeschirmt.

Dabei muss jedoch für mehrdimensionale Modelle noch geklärt werden, wann ein Punkt zwischen zwei anderen liegt. Nach enger Definition ist dies nur der Fall, wenn dieser Punkt auf der direkten Linie zwischen den beiden anderen Punkten liegt. Da dieser Fall jedoch selten eintritt, muss der Begriff in dieser Situation weiter gefasst und eine Abweichung zugelassen werden. Ein Punkt $x_2$ liegt zwischen zwei Punkten $x_1$ und $x_3$, wenn er nicht zu weit von der Strecke zwischen $x_1$ und $x_3$ abweicht. Die erlaubte Abweichung soll dabei mit steigender Distanz zwischen den Punkten zunehmen (siehe Abbildung 7.2). Sei $s_{i,j}$ die Strecke mit den Endpunkten $x_i$ und $x_j$.

| Variable | Beschreibung |
|---|---|
| $k$ | Anzahl der Dimensionen |
| $p_{min}$ | k-Vektor der minimalen Parameterwerte |
| $p_{max}$ | k-Vektor der maximalen Parameterwerte |
| $x_i$ | $i$-ter bekannter Punkt |
| $m$ | Anzahl der lokalen Kriging Metamodelle |
| $M$ | $m$-Vektor mit der Anzahl der Punkte je Cluster |
| $K$ | $m+1$-Vektor der Kriging Metamodelle<br>(Für Reduzierte Kriging Metamodelle gilt $m = 0$) |
| $K_0$ | Globales Metamodell |
| $K_1 \dots K_m$ | $m$ lokale Kriging Metamodelle |
| $\theta_i, \rho_i$ | Konfigurationsvektoren $\theta$ und $\rho$ des $i$-ten Kriging Metamodells |
| $K_{i,j}$ | Punkt $j$ des Modells $i$ |
| $\hat{y}_K(x)$ | Schätzwert des Kriging Metamodells $K$ für den Punkt $x$ |
| $s_i^2(x)$ | $s^2(x)$ des Kriging Metamodells $K_i$ |
| $f_{K,min}$ | $min\{f(x)\|x \in K\}$ |

Tabelle 7.1.: Hierarchisches Kriging: Übersicht der wichtigsten Variablen

**Definition 17 (Ein Punkt liegt bzgl. $\gamma$ zwischen zwei Punkten)**
*Ein Punkt $x_2$ liegt genau dann zwischen den Punkten $x_1$ und $x_3$, wenn der innere Winkel zwischen den Strecken $s_{1,2}$ und $s_{1,3}$ sowie der innere Winkel zwischen den Strecken $s_{2,3}$ und $s_{2,1}$ jeweils kleiner als der Winkel $\gamma$ ist.*

Bei dieser Heuristik ist zu beachten, dass die dominierten Punkte in absteigender bzw. aufsteigender Reihenfolge ihrer Funktionswerte auf Abschirmung gegenüber von oben bzw. von unten dominierenden Punkten überprüft werden, um die Anzahl der hinzugefügten Punkte zu begrenzen.

## 7.2. Hierarchische Kriging Metamodelle

Hierarchische Kriging Metamodelle erweitern den Ansatz der reduzierten Kriging Metamodelle, indem die dominierten Punkte in lokalen Metamodellen verwendet werden. Hierzu wird für jeden dominierenden Punkt ein Cluster gebildet, dem die nächstgelegenen dominierten Punkte hinzugefügt werden (siehe Abbildung 7.3). Für alle lokalen und den globalen Cluster werden Kriging Metamodelle erstellt. Die Approximationen dieser Modelle werden zu einer Approximation - dem hierarchischen Kriging Metamodell - verschmolzen (siehe Abbildung 7.4).

Um diese Metamodelle beschreiben zu können, müssen die bisherigen Schreibweisen erweitert werden. Zur besseren Übersicht wurden die wichtigsten Variablen in

Tabelle 7.1 zusammengefasst. Darüber hinaus müssen einige Formeln der normalen Kriging Metamodelle angepasst werden, damit sowohl das globale Metamodell als auch die lokalen Metamodelle berücksichtigt werden. Diese Anpassungen werden in den folgenden Abschnitten beschrieben.

Die Kriging-Distanz zweier Punkte ist abhängig von $\rho$ und $\theta$ des jeweiligen Metamodells $K_i$. Daher wird die Funktion *distance* zur Bestimmung der Kriging-Distanz um den Index $K_i$ erweitert. Die Kriging-Distanz zweier Punkte $x_1$ und $x_2$ bzgl. eines Kriging Metamodells $K_i$ berechnet sich dementsprechend als:

$$distance_{K_i}(x_1, x_2) = \sum_{j=1}^{k} \theta_{i,j} |x_{1,j} - x_{2,j}|^{\rho_{i,j}} \tag{7.2}$$

Analog zur Schreibweise der Distanz werden weitere Funktionen erweitert. Der erwartete Fehler an der Stelle eines Punkts $x$ bzgl. des Kriging Metamodells $K_i$ ist definiert als:

$$fehler_{K_i}(x) = s_i^2(x) \tag{7.3}$$

Der Distanz-Vektor eines Punkts $x$ bzgl. eines Kriging Metamodells $K_i$ ergibt sich als:

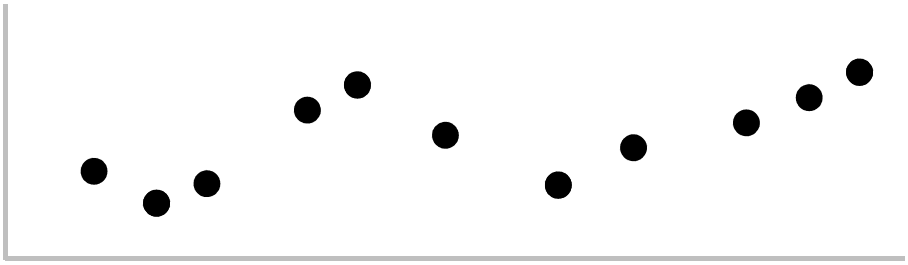
$$Distance_{K_i}(x) = (distance_{K_i}(x, K_{i,1}), \ldots, distance_{K_i}(x, K_{i,M_i}))' \tag{7.4}$$

Die Ausdehnung eines Kriging Metamodells $K_i$ berechnet sich als:

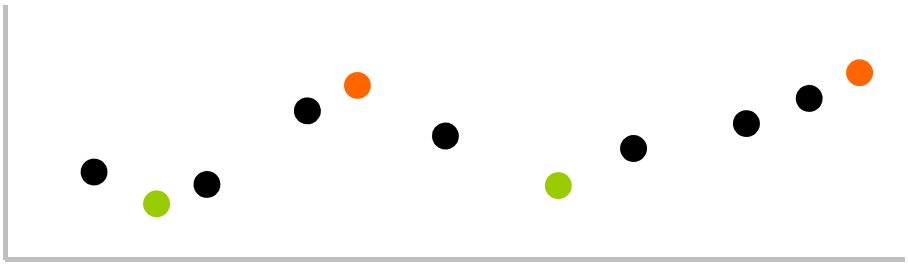
$$range(K_i) = max(Distance_{K_i}(K_{i,1})) \tag{7.5}$$

Das Expected Improvement (EI) eines hierarchischen Kriging Metamodells ist analog zum EI eines normalen Kriging Metamodells definiert. Dabei ist jedoch zu beachten, dass die Funktion $s_K(x)$ nicht der Funktion $s(x)$ eines normalen Kriging Metamodells entsprechen muss.

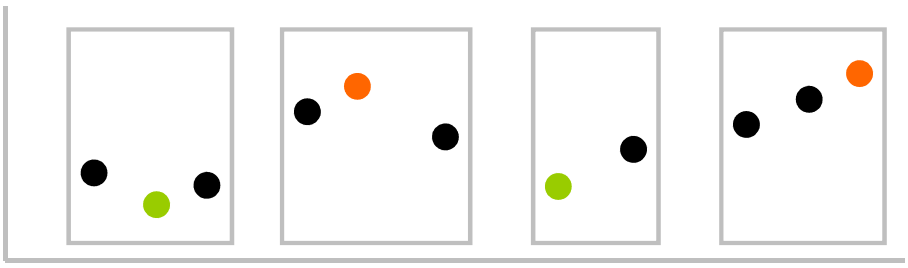
$$EI_K(x) = s_K(x)(u\Phi(u) + \phi(u)) \quad \text{mit} \quad u = \frac{f_{K,min} - \hat{y}_K(x)}{s_K(x)} \tag{7.6}$$



Punkte in einem Kriging Metamodell

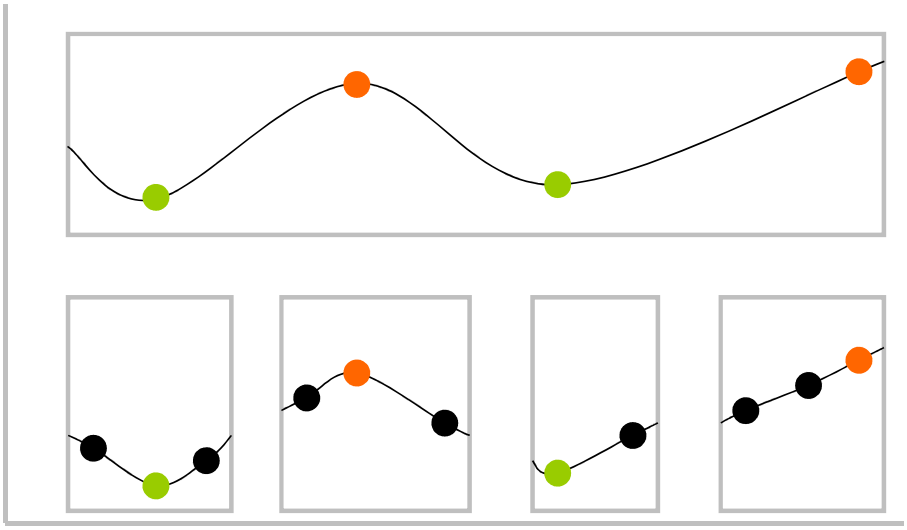


Identifikation dominanter Punkte
(grün für untere dominante Punkte und rot für obere dominante Punkte)

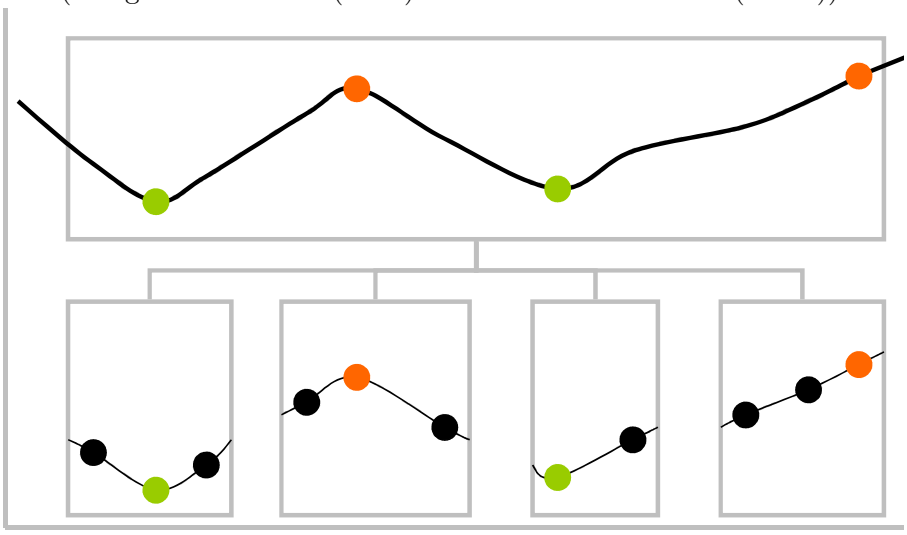


Clusterbildung entsprechend der dominanten Punkte

Abbildung 7.3.: Erstellung der Cluster eines hierarchischen Kriging Metamodells



Erstellung der einzelnen Metamodelle
(Ein globales Modell (oben) und vier lokale Modelle (unten))



Kombination der Metamodelle zu einem hierarchischen Metamodell

Abbildung 7.4.: Verschmelzung der lokalen Metamodelle mit dem globalen Metamodell zu einem hierarchischen Kriging Metamodell

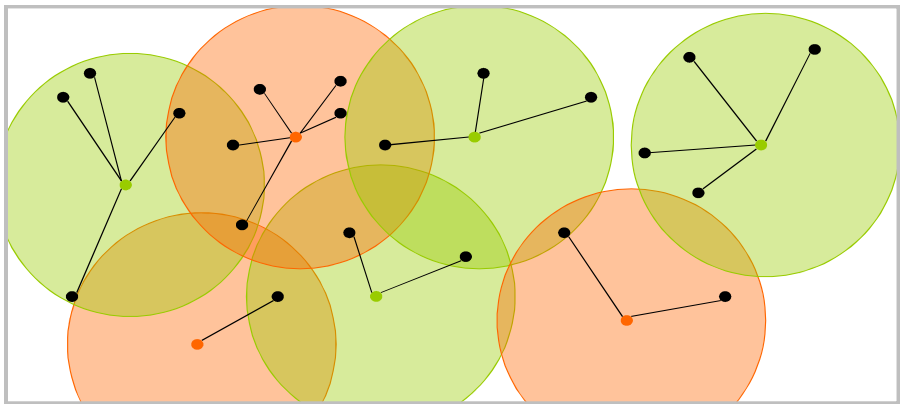


Abbildung 7.5.: Distanzbasierte Clusterbildung in hierarchischen Kriging Metamodellen für $k = 2$ Dimensionen mit dominanten Punkten.

### 7.2.1. Clusterbildung

Bisher wurde behandelt, wie verschiedene Werte eines hierarchischen Kriging Metamodells berechnet werden. Um diese Berechnungen durchführen zu können, muss zuerst ein hierarchisches Kriging Metamodell erstellt werden. Die folgenden Abschnitte befassen sich daher mit einem Verfahren, dass eine sinnvolle Gruppierung der Punkte ermöglicht. Diese Gruppierung ist dann die Grundlage für ein hierarchisches Kriging Metamodell. Der Ablauf des Verfahrens orientiert sich an Teilen der Clusteranalyse aus [BEPW03].

Aus den $n$ bekannten Punkten werden die $G_{r,s}$ Punkte gewählt, die alle anderen Punkte in einem Radius $r$ dominieren. Die dominierten Punkte werden anschließend den dominierenden Punkte zugeordnet und die entsprechenden Cluster gebildet (siehe Abbildung 7.5 und Pseudocode 7.1).

Hier werden zwei Ansätze für die Bestimmung von $r$ untersucht. Der Radius und die Distanz der Punkte zueinander bezieht sich dabei auf einen auf $[-1; 1]^k$ normierten Suchraum. Im ersten Ansatz wird $r_{1,n}$ in Abhängigkeit von der Anzahl der Parameter $k$, der Anzahl der Punkte im KM $n$ und der Länge der Diagonale durch den Suchraum berechnet:

$$r_{1,n} = \frac{|2_k|}{\sqrt{n}} = \frac{2\sqrt{k}}{\sqrt{n}} \tag{7.7}$$

$2_k$ ist dabei ein $k$-Vektor, dessen Elemente gleich 2 sind. Im zweiten Ansatz wird der Radius $r_{2_c}$ durch eine Konstante $c$ festgelegt:

1: // Phase 1: Markiere die dominanten Punkte im Vektor $e$ mit Werten größer 0.
2: $e = 0_n$;
3: $f = 0$;
4: **for** $i = 1, \ldots n$ **do**
5: $y_{min} = \min_{j=1,\ldots,n} \{y(x_j) | r \geq |x_i - x_j|\}$;
6: $y_{max} = \max_{j=1,\ldots,n} \{y(x_j) | r \geq |x_i - x_j|\}$;
7: **if** $y_{min} \geq y(x_i)$ **or** $y_{max} \leq y(x_i)$ **then**
8: $f$++;
9: $e_i = f$;
10: **end if**
11: **end for**
12: // Phase 2: Gruppiere die Punkte entsprechend der dominanten Punkte.
13: Initialisiere einen $h$-Vektor von Listen $C$.
14: **for** $i = 1, \ldots, n$ **do**
15: **if** $e_i > 0$ **then**
16: Füge $x_i$ am Anfang von $C_{e_i}$ ein.
17: **else**
18: $j = \text{argmin}_{b:e_b>0} \{| x_i - x_b |\}$;
19: Füge $x_i$ am Ende von $C_{e_j}$ ein.
20: **end if**
21: **end for**
22: Die Listen in $C$ enthalten die Punkte der jeweiligen Cluster mit dem zugehörigen dominanten Punkt an erster Stelle.

**Algorithmus 7.1:** Pseudocode zur Clusterbildung in hierarchischen KM

$$r_{2,c} = c \cdot |2_k| = 2c\sqrt{k} \tag{7.8}$$

Mit $r_{1,n}$ und $r_{2,c}$ kann das Clusteringverfahren durchgeführt werden. Die so bestimmten $G_{r,s}$ Punkte bilden den globalen Cluster und damit das Metamodell $K_0$. Für jeden dieser Punkte wird ein weiteres Metamodell erstellt, dass dem Index $j$ des jeweiligen Punkts im globalen Metamodell entsprechend $K_j$ benannt ist. Die restlichen $(G_{r,s} - n)$ Punkte werden jeweils dem Cluster $K_j$ zugeordnet, bei dem die Distanz zum j-ten Punkt im globalen Metamodell $K_{0,j}$ am Geringsten ist.

#### Probleme bei der Clusterbildung

Durch die Aufteilung der bekannten Punkte in ein globales und $m$ lokale Metamodelle müssen Funktionen entwickelt werden, die die Schätzwerte der einzelnen Teilmodelle möglichst gut kombinieren. Dabei ist nicht auszuschließen, dass je nach verwendeter Funktion zur Kombination der einzelnen Erwartungswerte ggf. einige

der den normalen Kriging Metamodellen zu Grunde liegenden Voraussetzungen verletzt werden. Daher müssen diese Kombinationsmöglichkeiten entweder verworfen oder auf die veränderte Situation Rücksicht genommen werden. Aus diesem Grund werden nun zuerst verschiedene Arten der Kombination der Erwartungswerte vorgestellt und anschließend Möglichkeiten zur Vermeidung derartiger Probleme vorgestellt.

### 7.2.2. Berechnung der kombinierten Erwartungswerte

Bei der Kombination der Metamodelle müssen verschiedene Funktionen berücksichtigt werden, zu denen z.B. das Expected Improvement $EI$ und der Standardfehler $s$ gehören. Da die meisten Funktionen auf der Berechnung des Standardfehlers $s$ beruhen, genügt es, diese Funktion anzupassen. Der Standardfehler eines hierarchischen Kriging Metamodells berechnet sich als:

$$s_K(x) = \frac{\sum_{i=0}^{m} a_i(x) * s_{K_i}(x)}{\sum_{i=0}^{m} a_i(x)} \tag{7.9}$$

Der Vektor $a(x) = (a_1(x), \ldots, a_m(x))'$ bestimmt den Einfluss der einzelnen Untermodelle und ist von der verwendeten Konfiguration abhängig. Die möglichen Konfigurationen werden im folgenden beschrieben.

Grundsätzlich werden die drei Typen **Reduziert**, **Submodell** und **Vollständig** unterschieden. Bei der Betrachtung eines **reduzierten** Kriging Metamodells werden nur die Punkte im globalen Metamodell $K_0$ berücksichtigt. Es gilt also $a_0(x) \geq 0$ und $a_i(x) = 0$ mit $i = 1, \ldots, m$. Der Typ **Submodell** betrachtet ausschließlich die Submodelle $K_i$ mit $i = 1, \ldots, m$. Es gilt also $a_0(x) = 0$ und $a_i(x) \geq 0$ mit $i = 1, \ldots, m$. Der Typ **Vollständig** betrachtet sowohl das globale als auch die Submodelle, daher gilt $a_i(x) \geq 0$ mit $i = 0, \ldots, m$.

Die konkreten Werte von $a(x)$ werden mittels einer Einflussfunktion berechnet. Da insbesondere die lokalen Metamodelle in großen Bereichen des Suchraums extrapolieren müssen und die so ermittelten Schätzwerte ungenau werden, ist es sinnvoll, den Einfluss eines lokalen Metamodells abhängig von der Distanz des Punkts $x$ zum lokalen Metamodell zu berechnen. Für diese Berechnungen stehen die fünf Funktionen **Center**, **Point**, **Range 1**, **Range 2** und **Range 3** zur Auswahl (siehe Abbildung 7.6). Die Berechnung von $a(x)$ für einen Punkt $x$ ergibt sich wie folgt:

**Center:** Der Einfluss des Metamodells hängt ausschließlich von der Distanz des Punkts $x$ vom Zentrum des Kriging Metamodells ab.

$$a_{1,i}(x) = \frac{1}{distance_{K_i}(K_{i,1}, x)} \tag{7.10}$$

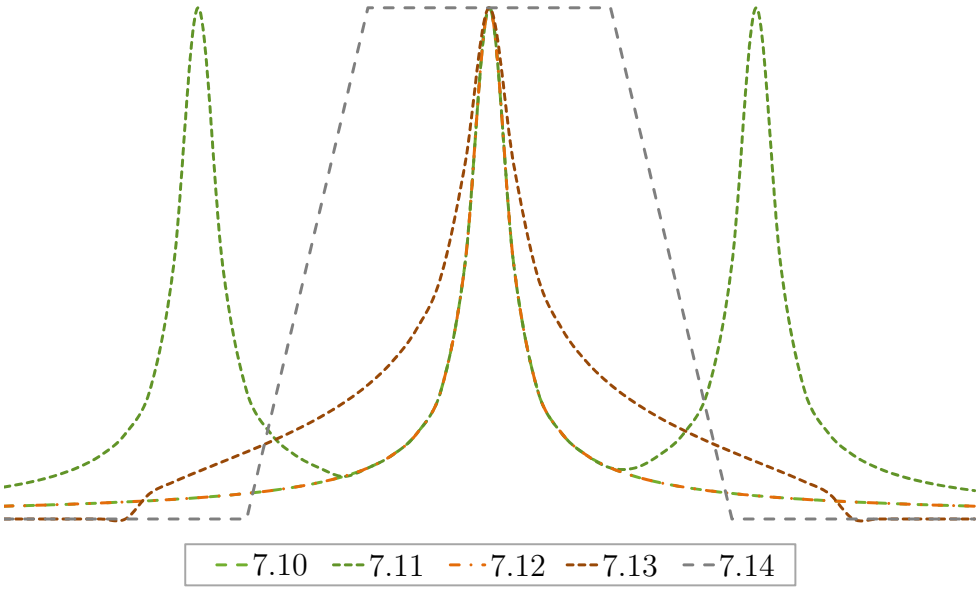



Abbildung 7.6.: Einheitlich skalierter Verlauf der Einflussfunktionen hierarchischer Kriging Metamodelle entsprechend Formeln 7.10 bis 7.14

**Point:** Der Einfluss des Metamodells hängt ausschließlich von der minimalen Distanz des Punkts $x$ zu einem Punkt aus dem Kriging Metamodell ab.

$$a_{2,i}(x) = \frac{1}{min(Distance_{K_i}(x))} \tag{7.11}$$

**Range 1:** Der Einfluss des Metamodells wird analog zu **Center** berechnet. Zusätzlich wird die Größe des Kriging Metamodells berücksichtigt.

$$a_{3,i}(x) = \frac{range(K_i)}{distance_{K_i}(K_{i,1}, x)} \tag{7.12}$$

**Range 2:** Der Einfluss wird in Anlehnung zu **Range 1** berechnet und zusätzlich so modifiziert, dass der Einfluss außerhalb der Ausdehnung des Kriging Metamodells gleich 0 ist.

$$a_{4,i}(x) = \sqrt{max\left(0, \frac{range(K_i)}{distance_{K_i}(K_{i,1}, x)} - 1\right)} \tag{7.13}$$

**Range 3:** Der Einfluss wird in Anlehnung zu **Range 2** berechnet. Dabei wird er jedoch so modifiziert, dass der Einfluss zwischen 0 und 1 liegt.

$$a_{5,i}(x) = \begin{cases} max\left(0, 1 - \sum_{i=1}^{m} a_i(x)\right) & ; \quad i = 0 \\ min\left(1, max\left(0, 2 - \frac{distance_{K_i}(K_{i,1}, x)}{range(K_i)}\right)\right) & ; \quad \text{sonst} \end{cases} \tag{7.14}$$

### 7.2.3. Vermeidung kleiner Cluster

Die bisherigen Beschreibungen gehen davon aus, dass die gebildeten Cluster ausreichend groß sind, um eine sinnvolle Approximation zu bilden. Besteht ein Cluster jedoch nur aus sehr wenigen Punkten, enthält dieser zu wenig Informationen, um auf dessen Basis verlässliche Werte berechnen zu können. Daher werden kleine Cluster aufgelöst, indem die enthaltenen Punkte mit Ausnahme des dominanten Punkts in den jeweils nächstgelegenen Cluster verschoben werden. Der dominante Punkt des ehemaligen Clusters verbleibt ausschließlich im globalen Metamodell. Der Grenzwert zur Auflösung kleiner Cluster wurde auf die Anzahl der Dimensionen des Modells $k$ festgelegt.

### 7.2.4. Neubildung von Clustern

Da im Verlaufe der Optimierung durch die Auswertung neuer Punkte zusätzliche Informationen in das Metamodell eingehen, muss ggf. die Aufteilung der Cluster neu berechnet werden. Da die Berechnung der Aufteilung auf die Cluster jedoch aufwändig ist, sollte dies nicht bei jedem neu hinzugefügten Punkt durchgeführt werden. Zu diesem Zweck müssen Heuristiken entwickelt werden, die den Zeitpunkt der Neustrukturierung bestimmen. Grundsätzlich gilt bei allen Heuristiken, dass die neu hinzugefügten Punkte zuerst in das globale Metamodell eingefügt werden. Diese Punkte werden erst in einen anderen Cluster aufgenommen, wenn Sie entweder von der Heuristik explizit einem solchen zugeordnet werden oder eine vollständige Neuberechnung der Cluster durchgeführt wird.

**Heuristik 1:** Bei der ersten Heuristik werden alle $a$ Schritte die Cluster neu bestimmt. Abhängig von der Wahl von $a$ wird das Metamodell häufiger neustrukturiert oder dem Verfahren die Möglichkeit gegeben, sich anhand der aktuellen Struktur ergebende Minima detaillierter zu betrachten.

**Heuristik 2:** Die zweite Heuristik führt dann eine Neustrukturierung durch, wenn keine weitere Verbesserung zu erwarten ist, dazu wird überprüft, ob das aktuelle Metamodell in $a$ aufeinander folgenden Schritten keine weitere Verbesserung ergeben hat.

**Heuristik 3:** Die dritte Heuristik berücksichtigt die Veränderung der Cluster bei der Entscheidung der Neubildung. Dabei wird die Anzahl $m_c$ der Cluster gezählt, bei denen der Mittelpunkt nicht gleichzeitig der Punkt mit dem besten Messwert ist. Es werden also die Cluster berücksichtigt, bei denen nach dem Clusterbildung ein besserer Punkt hinzugefügt wurde. Nach jedem Kriging-Schritt wird dabei auf der Basis einer gleichverteilten Zufallsvariable entschieden, ob die Cluster neu gebildet werden sollen. Die Wahrscheinlichkeit beträgt $\frac{m_c+1}{m+1}$. Die zusätzlich addierte 1 soll dabei sicherstellen, dass jederzeit eine Neubildung möglich ist.

**Heuristik 4:** Die vierte Heuristik berücksichtigt das Verhältnis der Punkte im globalen Metamodell zur durchschnittlichen Anzahl der Punkte in den Clustern. Entsprechend der Formel $|K_0| - m > \frac{n}{m}$ werden die Cluster genau dann neu gebildet, wenn in das globale Metamodell mehr Punkte eingefügt wurden, als durchschnittlich in den lokalen Metamodellen vorhanden sind. Grundsätzlich soll dadurch vermieden werden, dass das globale Metamodell zu stark anwächst und so die Rechenzeit für die Anpassung des globalen Metamodells zu stark ansteigt.

**Heuristik 5:** Diese Heuristik baut auf der Heuristik 4 auf und verhindert zusätzlich, dass neue Punkte zu häufig in das globale Metamodell eingefügt werden. Dabei wird zuerst untersucht, ob der neu hinzugefügte Punkt $x$ innerhalb des Einflussbereichs eines lokalen Metamodells liegt. Dabei wird überprüft, ob die Distanz von $x$ zum dominanten Punkt des lokalen Metamodells geringer als die maximale Distanz des dominanten Punkts zu den Punkten des lokalen Metamodells ist. Sollte dies bei mehreren lokalen Metamodellen der Fall sein, so wird das lokale Metamodell genutzt, dessen Distanz vom dominanten Punkt zu $x$ am geringsten ist. In das so ausgewählte Metamodell wird der Punkt anschließend eingefügt. Wird kein Metamodell ausgewählt, so wird $x$ dem globalen Metamodell hinzugefügt. In dem Fall, dass $x$ einem lokalen Metamodell hinzugefügt wird, wird darüber hinaus überprüft, ob $x$ den bisher dominanten Punkt des lokalen Metamodells dominiert. Ist dies der Fall übernimmt der Punkt $x$ dessen Position im lokalen und globalen Metamodell.

Auf diese Weise sollen unnötige Neuberechnungen der Cluster vermieden und die neuen Punkte sinnvoll in das Metamodell integriert werden. Die Struktur des Metamodells wird dabei angepasst, so dass die zu Grunde liegenden Eigenschaften der hierarchischen Kriging Metamodelle erhalten bleiben.

### 7.2.5. Erweiterte Information in Clustern

Zusätzlich zu den bisher beschriebenen Informationen können die Cluster mit weiteren Informationen bestückt werden. An dieser Stelle werden zwei Varianten beschrieben. Die erste erweitert das globale Kriging Metamodell $K_0$, so dass alle bekannten Punkte enthalten sind. Es handelt sich dann bei $K_0$ um ein normales Kriging Metamodell, dass durch die lokalen Metamodelle an den Extrema unterstützt wird. Die

| **Zeichen** | **A** | **B** | **C** | **D** |
|---|---|---|---|---|
| **Kriterium** | Typ | Einfluss-funktion | Erweiterung des globalen Metamodells | Erweiterung des lokalen Metamodells |
| **Werte** | GLOBAL, SUB, ALL | 1, 2, 3, 4, 5 | 1, 0 | 1, 0 |
| **Bedeutung der Werte** | Nur globales Metamodell, nur Submodelle, kombiniertes Metamodell | *Center*, *Point*, *Range*1, *Range*2, *Range*3 | ja, nein | ja, nein |

Tabelle 7.2.: Namensschema der Testläufe für das hierarchische Kriging

zweite Variante erweitert die lokalen Kriging Metamodelle um alle Punkte des ursprünglichen globalen Kriging Metamodells $K_0$. Damit wird die Extrapolation in den lokalen Metamodellen reduziert. Beide Erweiterungen haben jedoch zur Folge, dass der Rechenaufwand steigt. Insbesondere bei der ersten Variante liegt der Rechenaufwand über der normaler Kriging Metamodelle.

### 7.2.6. Anpassung der Kriging Metamodelle

Kriging Metamodelle werden üblicherweise mittels Maximum-Likelihood-Funktionen an die vorhandenen Information angepasst. Im Falle von hierarchischen Kriging-Metamodellen muss dies $m+1$ mal durchgeführt werden. Da die Informationen jedoch in bestimmtem Maße zusammenhängen, können hier ggf. Berechnungen eingespart werden. Ein Ansatz ist, die lokalen Metamodelle mit den Werten des globalen Metamodells zu konfigurieren, um so die Berechnungen für die lokalen Metamodelle nicht durchführen zu müssen.

## 7.3. Experimente und Ergebnisse

Die folgenden Abschnitte beschreiben zuerst den Aufbau und Ablauf der Experimente und führen einige Bezeichnungen ein. Anschließend werden die Ergebnisse vorgestellt und diskutiert.

Die Experimente werden wie folgt durchgeführt: Für jede Funktion werden 32 Konfigurationen der hierarchischen Kriging Metamodelle erstellt. Die 32 Konfigurationen entsprechen den sinnvollen Kombination der Typen, der Einflussfunktionen und der Erweiterungen sowie dem normalen KM. Die Benennung der Konfigurationen folgt dem Muster „A-B-C-D“ (siehe Tabelle 7.2). „A“ steht für den Typ, „B“ für die Einflussfunktion, „C“ und „D“ zeigen an, ob das globale Metamodell bzw. die lokalen

Metamodelle erweitert werden. Im Bereich Typ (A) steht „GLOBAL“ dafür, dass nur ein globales Metamodell erstellt wird, „SUB“ bedeutet, dass nur die Submodelle berücksichtigt werden und „ALL“ kennzeichnet, dass sowohl das globale Metamodelle als auch Submodelle berücksichtigt werden. Die Ziffern im Bereich Einflussfunktionen (B) entsprechen den Ziffern der Einflussfunktionen aus Kapitel 7.2.2 und stehen von 1 aufsteigend für „Center“, „Point“, „Range1“, „Range2“ und „Range3“. Ein normales Kriging Metamodell kann in diesem Schema mit „GLOBAL-1-1-0“ bezeichnet werden. Für jede Funktion werden alle Experimente durchgeführt und 121 mal wiederholt.

### 7.3.1. Approximationsgüte

In diesem Abschnitt wird die Approximationsgüte der hierarchischen Kriging Metamodelle mit der Approximationsgüte der normalen Kriging Metamodelle verglichen. Dazu werden für jede Konfiguration A-B-C-D (siehe Tabelle 7.2) die entsprechenden Metamodelle für 100 unterschiedliche Latin Hypercube Samplings (siehe Kapitel 3.2.4) berechnet. Die Experimente werden mit LHS bestehend aus 60 bzw. 120 Punkten durchgeführt. Die Messung des realen und des durch das (hierarchische) Kriging Metamodell erwarteten Fehlers wird an 1000 zufällig ausgewählten Punkten durchgeführt.

In den Abbildungen 7.7 und 7.8 sind der durchschnittliche, absolute Fehler der Metamodelle mit 60 bzw. 120 Punkten dargestellt. In Abbildung 7.7 ist deutlich zu sehen, dass für die Benchmarkfunktionen Goldstein-Price und Ackley die hierarchischen Kriging Metamodelle zu geringfügig schlechteren Approximationen führen. Die Sinus-Funktion kann von normalen Kriging Metamodellen deutlich besser als mit hierarchischen Kriging Metamodellen approximiert werden. Im Fall der Schaffer-Funktion sind die hierarchischen Kriging Metamodellen den normalen Kriging Metamodellen überlegen. Diese Aussagen für (hierarchische) Kriging Metamodelle mit 60 Punkte gelten analog für (hierarchische) Kriging Metamodelle mit 120 Punkten (vgl. Abbildung 7.8).

In den Abbildungen 7.9 und 7.10 ist das durchschnittliche Verhältnis von realem Fehler im Metamodell und durch die Metamodelle erwartetem Fehler dargestellt. Im Idealfall ist das Verhältnis nicht nur im Durchschnitt sondern an jedem Punkt exakt 1. In diesem Fall entspricht der erwartete Fehler an jeder Stelle des Modells dem realen Fehler. Ist das durchschnittliche Verhältnis größer als 1, tendiert das Metamodell zu einer Überschätzung des realen Fehlers; im entgegengesetzten Fall wird der reale Fehler unterschätzt. Für eine Optimierung basierend auf solchen Metamodellen bedeutet dies, dass im ersten Fall für die Optimierung uninteressante Regionen möglicherweise untersucht werden bzw. im zweiten Fall Bereiche mit potentiell globalem Optimum nicht betrachtet werden. Auf eine Optimierung hat daher

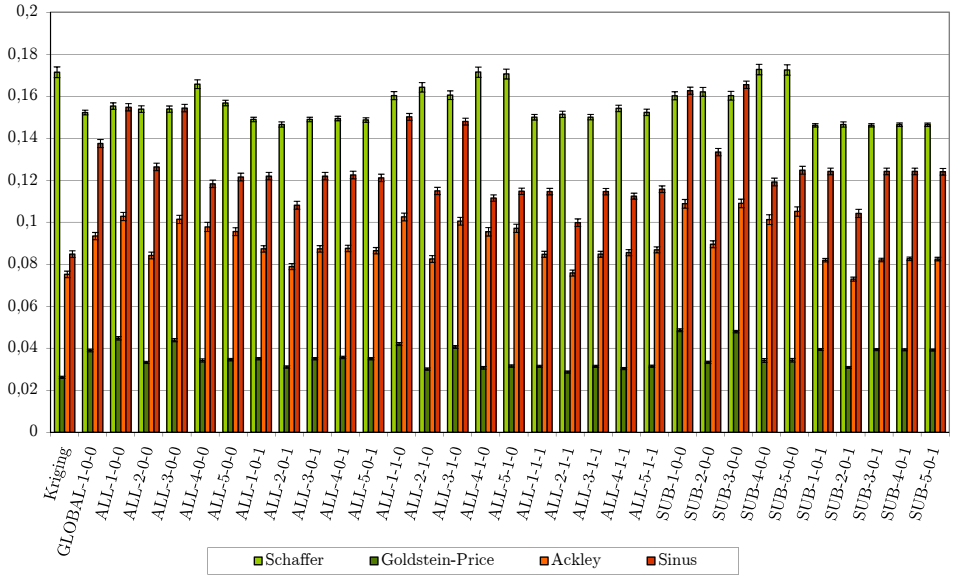



Abbildung 7.7.: Durchschnittlicher realer Fehler bei 60 Punkten im KM mit 90% Konfidenzintervall

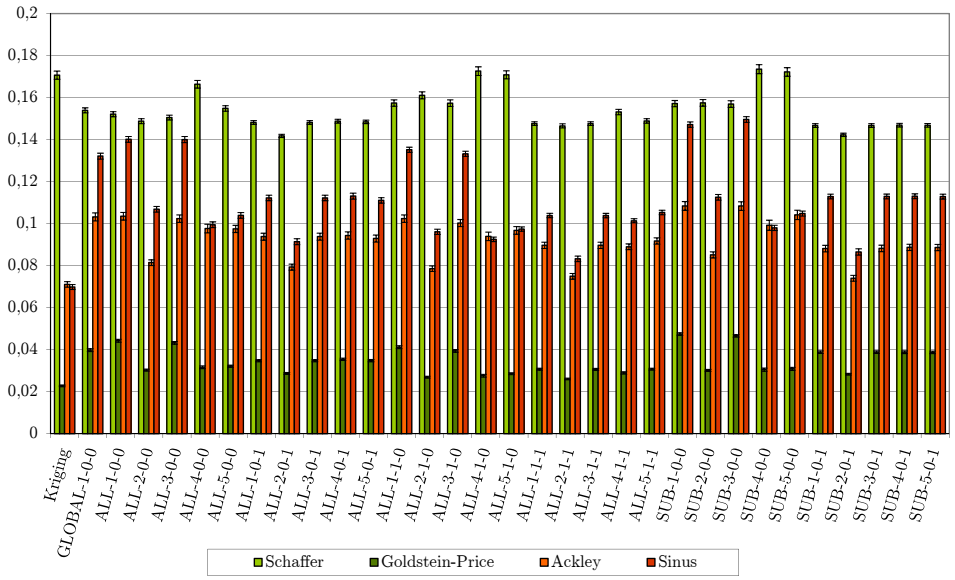



Abbildung 7.8.: Durchschnittlicher realer Fehler bei 120 Punkten im KM mit 90% Konfidenzintervall

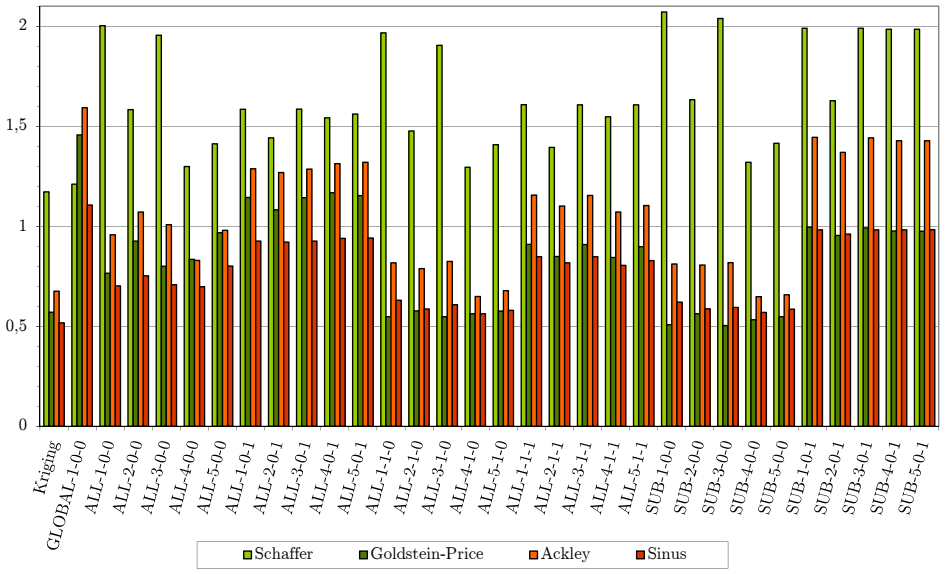



Abbildung 7.9.: Verhältnis von erwartetem und realem Fehler bei 60 Punkten im KM

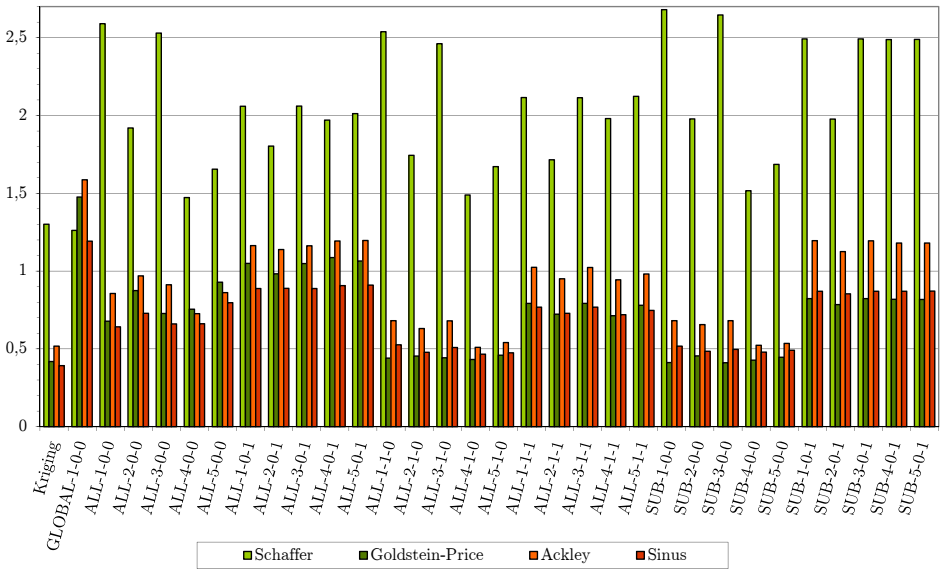



Abbildung 7.10.: Verhältnis von erwartetem und realem Fehler bei 120 Punkten im KM

der zweite Fall stärkere Auswirkungen als der erste Fall. Daher ist eine Überschätzung des realen Fehlers positiver als eine Unterschätzung zu bewerten. Die Abbildungen 7.9 und 7.10 zeigen deutlich, dass das Verhältnis von erwartetem zu realem Fehler bei den hierarchischen Kriging Metamodellen in allen Fällen mindestens genauso groß wie das entsprechende Verhältnis bei normalen Kriging Metamodellen ist. Sowohl bei der Goldstein-Price- als auch bei der Sinus-Funktion wird von vielen hierarchischen Kriging Metamodellen ein deutlich realistischerer Fehler erwartet. Bei der Ackley-Funktion wird der reale Fehler moderat überschätzt. Nur im Fall der Schaffer-Funktion wird der reale Fehler teils deutlich überschätzt.

Zusammenfassend kann festgestellt werden, dass hierarchische Kriging Metamodelle eine ähnliche Approximationsgüte im Vergleich zu normalen Kriging Metamodellen aufweisen. Darüber hinaus bieten sie in vielen Fällen eine für die Optimierung besser geeignete Einschätzung des Modellfehlers.

### 7.3.2. Zeitbedarf und Ergebnisgüte bei der Optimierung

Der zweite Teil der Experimente untersucht sowohl den Zeitbedarf der Optimierung mit hierarchischen Kriging Metamodellen als auch die Güte der Ergebnisse. Dazu wird für jede Konfiguration (siehe Tabelle 7.2) KMO mit 100 Replikationen durchgeführt und der Durchschnitt der jeweiligen Werte berechnet. In den Abbildungen 7.11 und 7.12 werden jeweils die Durchschnittswerte aller Replikationen für jede Methode in Relation zu den Ergebnissen des normalen Kriging Metamodells aufgeführt. Einer der Gründe für die Einführung hierarchischer Kriging Metamodelle war der stark steigende Zeitaufwand während der Optimierung (Kapitel 6.1.3). Dieser kann beim Einsatz normaler Kriging Metamodelle bereits bei 120 Auswertungen einige Minuten betragen. Dieser Zeitaufwand kann durch den Einsatz hierarchischer Metamodelle deutlich reduziert werden.

Es ist jedoch nicht sinnvoll, nur den Zeitaufwand zu betrachten. Die benötigte Zeit muss immer in Verbindung mit der erreichten Güte der Ergebnisse stehen. Tabelle 7.3 zeigt die Ergebnisse in kombinierter Form. Für jede Konfiguration enthält die Tabelle ein Feld mit zwei Zeichen. Das erste Zeichen bezieht sich auf den Zeitbedarf und das zweite auf die Güte der Ergebnisse. Dabei steht ein + für im Durchschnitt bessere Ergebnisse als das normale Kriging-Verfahren und ein - für schlechtere Werte. Daraus ergibt sich, dass die Konfiguration ALL-2-0-0 und die Konfigurationen ALL-$x$-0-1 für $x = 1, \ldots, 5$ den normalen Kriging Metamodellen in vielen Fällen überlegen sind und die restlichen Konfigurationen entweder im Zeitbedarf oder in der Ergebnisgüte bessere Ergebnisse liefern als das normale Kriging-Verfahren.

Aufbauend auf der vorgestellten Experimentserie wurden vier weitere Experimentserien durchgeführt. Dabei wurde die jeweils eingesetzte Funktion mit einer normalverteilten Störung versehen. Die Standardabweichung der Störung für die Experimentserien steigt an und beträgt $0,001$, $0,01$, $0,05$ und $0,1$. In den Abbildungen 7.13,

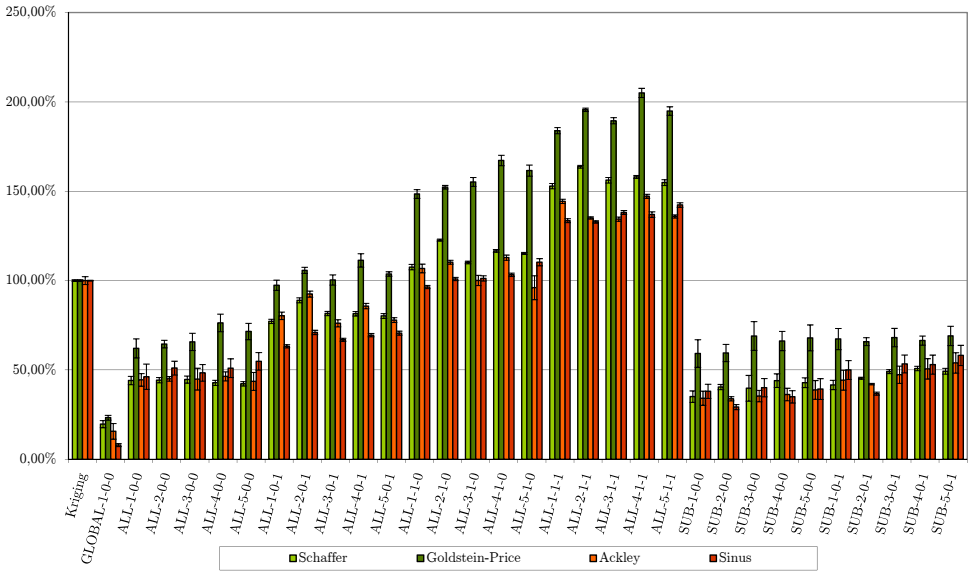



Abbildung 7.11.: Zeitbedarf für eine Optimierung mit 120 Auswertungen relativ zu normalem Kriging mit 90% Konfidenzintervall

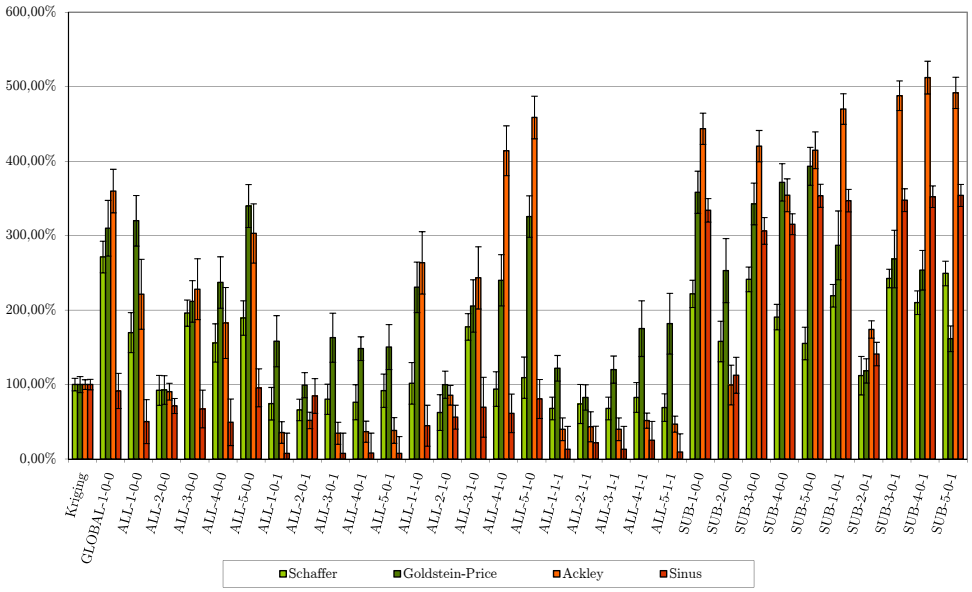



Abbildung 7.12.: Distanz zum Optimum bei 120 Auswertungen relativ zu normalen Kriging mit 90% Konfidenzintervall

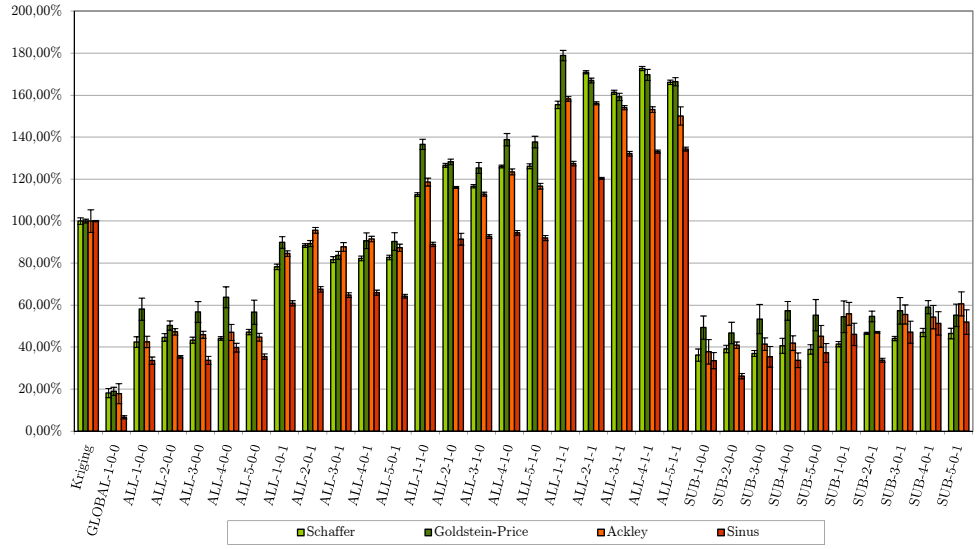



Abbildung 7.13.: Zeitbedarf für eine Optimierung mit 120 Auswertungen relativ zu normalem Kriging bei einer normalverteilten Störung mit $\sigma^2 = 0,001$ und 90% Konfidenzintervall

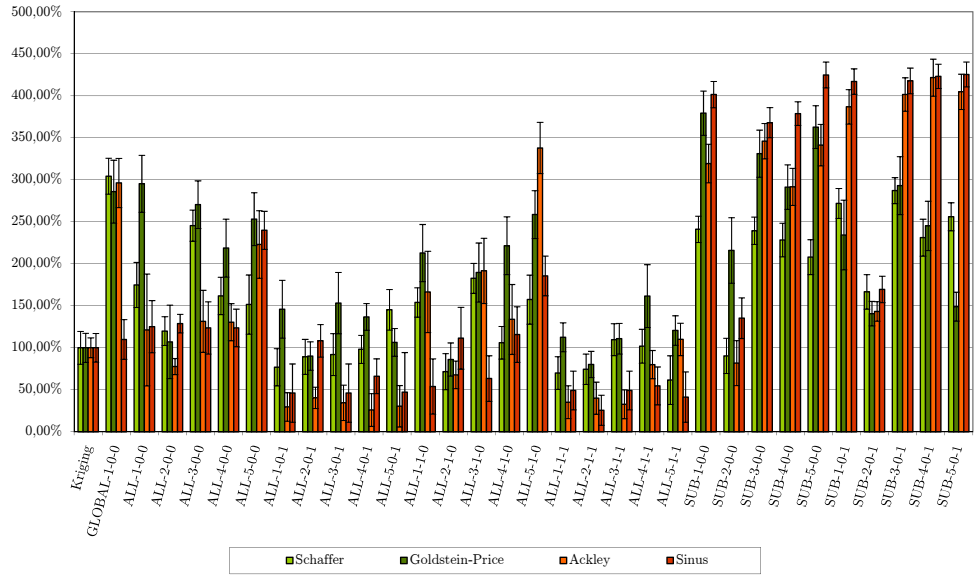



Abbildung 7.14.: Distanz zum Optimum bei 120 Auswertungen relativ zu normalem Kriging bei einer normalverteilten Störung mit $\sigma^2 = 0,001$ und 90% Konfidenzintervall

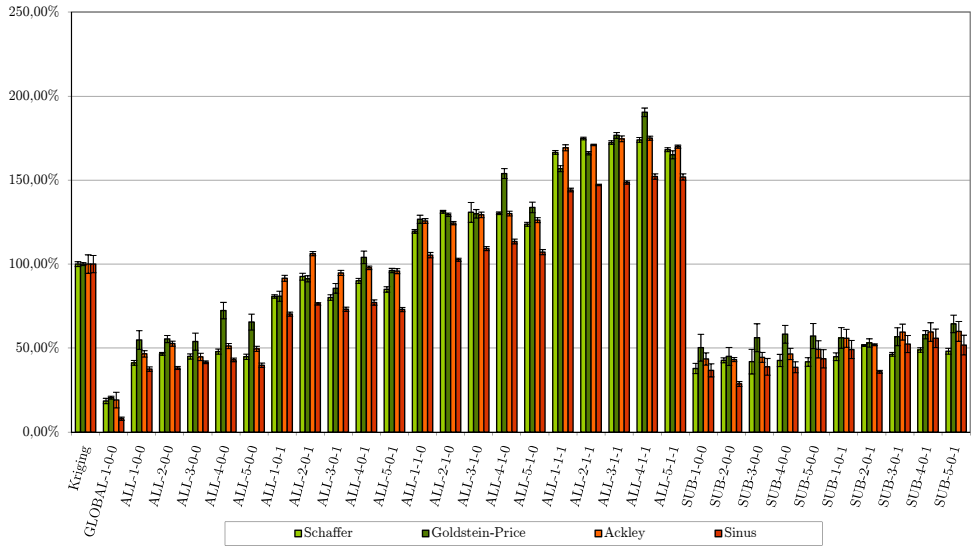



Abbildung 7.15.: Zeitbedarf für eine Optimierung mit 120 Auswertungen relativ zu normalem Kriging bei einer normalverteilten Störung mit $\sigma^2 = 0,01$ und 90% Konfidenzintervall

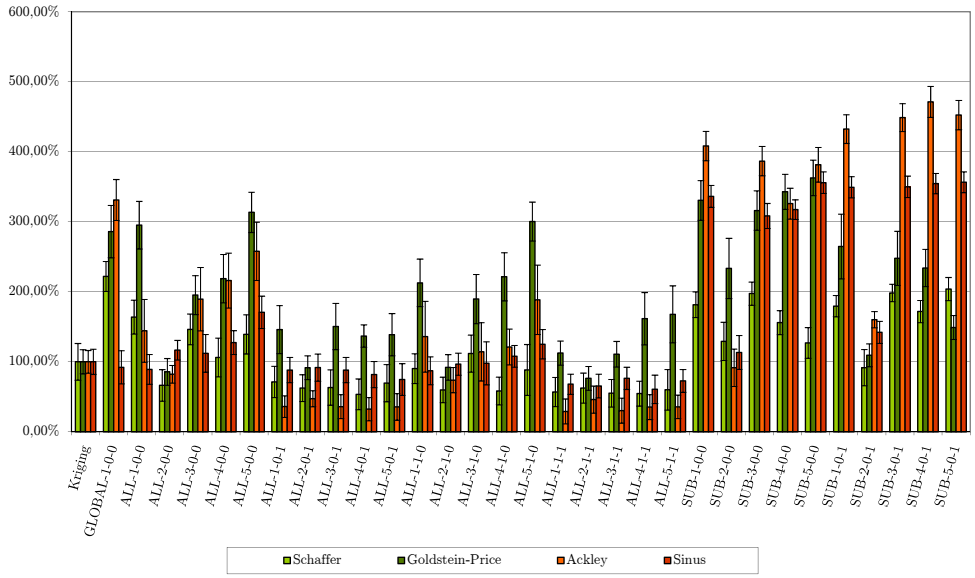



Abbildung 7.16.: Distanz zum Optimum bei 120 Auswertungen relativ zu normalem Kriging bei einer normalverteilten Störung mit $\sigma^2 = 0,01$ und 90% Konfidenzintervall

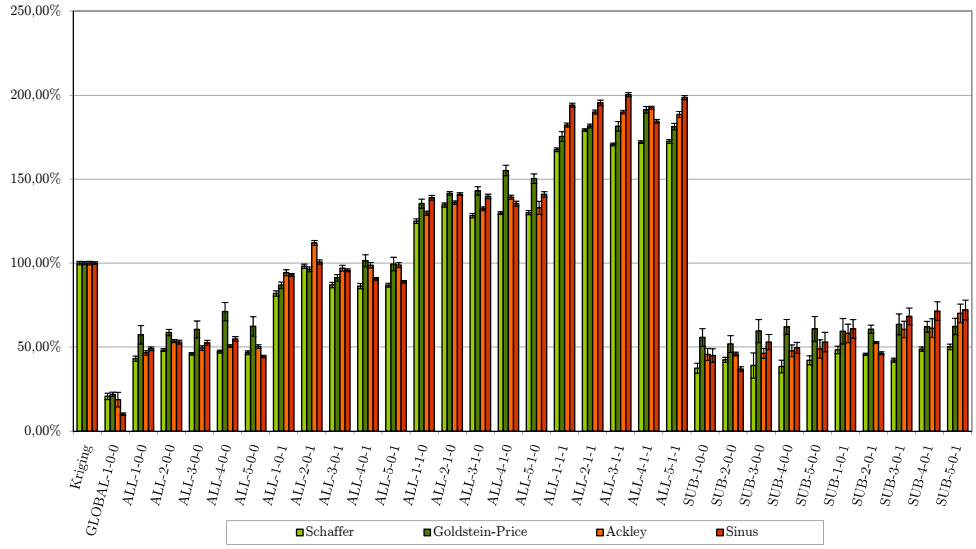



Abbildung 7.17.: Zeitbedarf für eine Optimierung mit 120 Auswertungen relativ zu normalem Kriging bei einer normalverteilten Störung mit $\sigma^2 = 0,05$ und 90% Konfidenzintervall

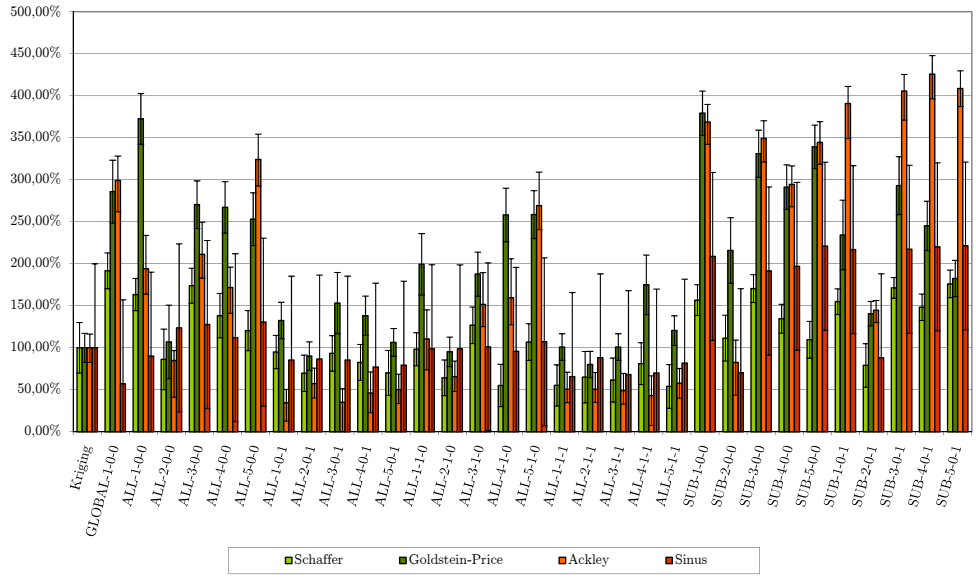



Abbildung 7.18.: Distanz zum Optimum bei 120 Auswertungen relativ zu normalem Kriging bei einer normalverteilten Störung mit $\sigma^2 = 0,05$ und 90% Konfidenzintervall

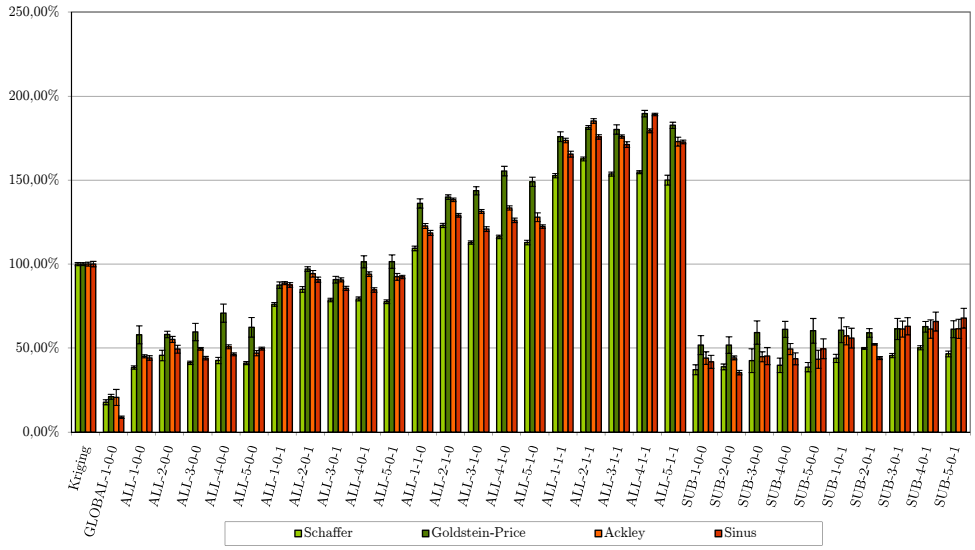



Abbildung 7.19.: Zeitbedarf für eine Optimierung mit 120 Auswertungen relativ zu normalem Kriging bei einer normalverteilten Störung mit $\sigma^2 = 0,1$ und 90% Konfidenzintervall

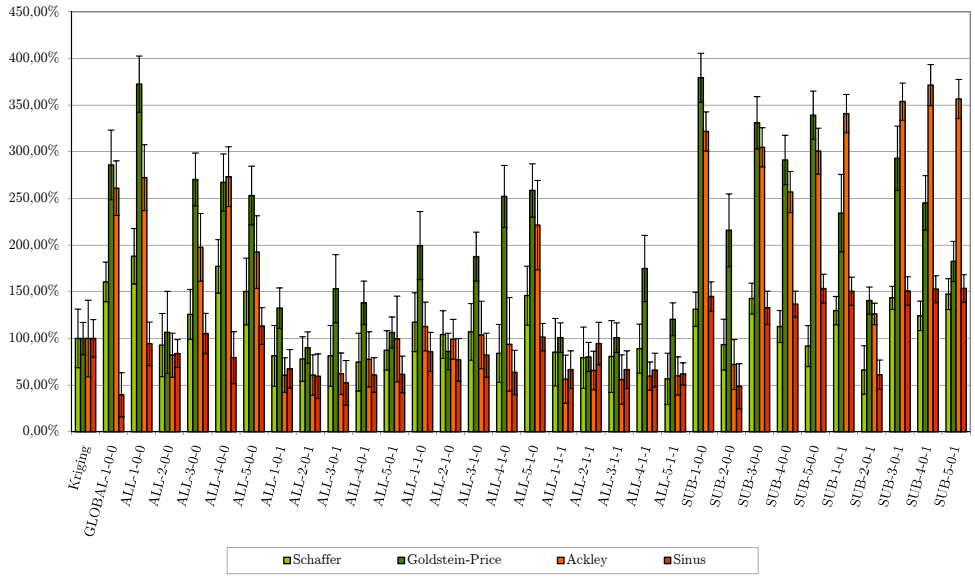



Abbildung 7.20.: Distanz zum Optimum bei 120 Auswertungen relativ zu normalem Kriging bei einer normalverteilten Störung mit $\sigma^2 = 0,1$ und 90% Konfidenzintervall

| Basis- | Erweiterung | | | |
|---|---|---|---|---|
| konfiguration | 0-0 | 0-1 | 1-0 | 1-1 |
| GLOBAL-1 | +- | | | |
| ALL-1 | +- | ++ | – | -+ |
| ALL-2 | ++ | ++ | -+ | -+ |
| ALL-3 | +- | ++ | – | -+ |
| ALL-4 | +- | ++ | – | -+ |
| ALL-5 | +- | ++ | – | -+ |
| SUB-1 | +- | +- | | |
| SUB-2 | +- | +- | | |
| SUB-3 | +- | +- | | |
| SUB-4 | +- | +- | | |
| SUB-5 | +- | +- | | |

Tabelle 7.3.: Bewertung der Ergebnisse zum hierarchischen Kriging

7.14, 7.15, 7.16, 7.17, 7.18, 7.19 und 7.20 sind die Ergebnisse analog zu den Abbildungen 7.11 und 7.12 dargestellt. Da die Ergebnisse relativ zu den Ergebnissen von KMO dargestellt werden, sind die Ergebnisse sehr ähnlich. Daraus ergibt sich, dass sich Störungen in Funktionen auf KMO und KMO mit hierarchischen Modellen mit gleicher Intensität auswirken. Hierarchische Kriging Metamodelle sind daher in diesem Aspekt genauso robust wie normale Kriging Metamodelle. Darüber hinaus lassen sich die aus der ersten Experimentserie gewonnenen Erkenntnisse direkt auf die restlichen Experimentserien übertragen.

## 7.4. Zusammenfassung

Hierarchische Kriging Metamodelle stellen eine gute Ergänzung der normalen Kriging Metamodelle dar. Sie ermöglichen, die Qualität der Optimierungsergebnisse zu erhöhen. Dabei wird jedoch die Rechenzeit ggf. weiter angehoben. In vielen Fällen ist aber selbst diese Steigerung der Rechenzeit im Vergleich zur Zeit, die z.B. für die durchzuführenden Simulationen benötigt wird, unerheblich. Gerade bei zeitintensiven Auswertungen sind hierarchische Kriging Metamodelle daher den normalen Kriging Metamodellen vorzuziehen.

Im Gegensatz dazu bieten reduzierte Kriging Metamodelle die Möglichkeit nur für das Metamodell wichtige Punkte im KM zu berücksichtigen. Durch dieses Verhalten wird zwar im Vergleich zu den normalen KM die Qualität der Approximation reduziert, die benötigte Rechenzeit aber ebenfalls signifikant verringert.

# 8. Hybride Kriging-Verfahren

In vielen Situationen können Kriging Metamodelle Funktionen mit einer geringen Anzahl bekannter Punkte gut approximieren. Auch die KMO hat gezeigt, dass für eine globale Optimierung in der Regel deutlich weniger Punkte betrachtet werden müssen, als dies bei vielen anderen globalen Suchheuristiken der Fall ist. Die Stärke dieses Verfahrens ist die die Identifikation von Bereichen, in denen wahrscheinlich ein (lokales) Optimum liegt.

Im Gegensatz dazu benötigt die Suche nach dem wirklichen Optimum eine relativ hohe Anzahl an Punkten, da die Annäherung an das Optimum häufig nur in kleinen Schritten erfolgt. Ist jedoch ein Bereich eines (lokalen) Optimums bereits identifiziert, kann dieser lokale Bereich als unimodale Funktion betrachtet werden. In diesem Kontext haben lokale Suchheuristiken ihre Stärken, so dass eine Kombination beider Verfahren untersuchenswert ist.

In diesem Kapitel wird daher eine Kombinationsmöglichkeit von KM mit dem lokalen Verfahren PS (siehe Kapitel 5.4) beschrieben und das neu entstandene Verfahren auf seine Einsetzbarkeit untersucht.

## 8.1. Kombination von Kriging Metamodellen und Pattern Search

Der hier vorgeschlagene Ansatz **hybride Kriging Metamodell basierte Optimierung** (HKO) basiert im wesentlichen auf der normalen KMO, wird aber um einige „lokale“ Teilschritte ergänzt. Grundsätzlich wird in jeder Iteration entsprechend der KMO ein Punkt mit maximalen EI gesucht und dem Metamodell hinzugefügt. So wird sicher gestellt, dass weiterhin eine globale Suche durchgeführt wird. Zusätzlich wird ein lokaler Inspektionsschritt eingefügt. Dieser arbeitet wie folgt:

Um einen Bereich zu finden, der möglicherweise ein Optimum enthält, werden die bekannten Punkte analog zu dem bereits in 7.2.1 beschriebenen Algorithmus untersucht. Von den potentiellen Minima wird dabei der Punkt ausgewählt, der in einer noch nicht untersuchten Region liegt und von den übrig gebliebenen Punkten den niedrigsten Funktionswert hat. Auf diese Weise wird sichergestellt, dass viel versprechende Bereiche zuerst untersucht werden, anschließend aber auch andere Bereiche berücksichtigt werden.

| Nr | Verfahren | $h_r$ | EI | $f$ oder KM |
|---|---|---|---|---|
| 1 | KMO | - | - | - |
| 2 | HKO | 0,05 | $EI_{hko}$ | $f$ |
| 3 | HKO | 0,05 | $EI_{hko}$ | KM |
| 4 | HKO | 0,05 | $EI$ | $f$ |
| 5 | HKO | 0,05 | $EI$ | KM |
| 6 | HKO | 0,10 | $EI_{hko}$ | $f$ |
| 7 | HKO | 0,10 | $EI_{hko}$ | KM |
| 8 | HKO | 0,10 | $EI$ | $f$ |
| 9 | HKO | 0,10 | $EI$ | KM |
| 10 | HKO | 0,20 | $EI_{hko}$ | $f$ |
| 11 | HKO | 0,20 | $EI_{hko}$ | KM |
| 12 | HKO | 0,20 | $EI$ | $f$ |
| 13 | HKO | 0,20 | $EI$ | KM |
| 14 | HKO | 0,30 | $EI_{hko}$ | $f$ |
| 15 | HKO | 0,30 | $EI_{hko}$ | KM |
| 16 | HKO | 0,30 | $EI$ | $f$ |
| 17 | HKO | 0,30 | $EI$ | KM |
| 18 | HKO | 0,50 | $EI_{hko}$ | $f$ |
| 19 | HKO | 0,50 | $EI_{hko}$ | KM |
| 20 | HKO | 0,50 | $EI$ | $f$ |
| 21 | HKO | 0,50 | $EI$ | KM |
| 22 | HKO | 0,80 | $EI_{hko}$ | $f$ |
| 23 | HKO | 0,80 | $EI_{hko}$ | KM |
| 24 | HKO | 0,80 | $EI$ | $f$ |
| 25 | HKO | 0,80 | $EI$ | KM |

Tabelle 8.1.: Übersicht der untersuchten Konfigurationen des HKO

Ist ein Punkt $x_i$ für die Inspektion ausgewählt worden, wird dieser als besucht markiert und $x_s$ genannt. Dadurch werden alle anderen Punkte, die innerhalb eines Radius $h_r$ um diesen Punkt liegen, bei späteren Inspektionsschritten nicht berücksichtigt. Beginnend mit dem Punkt $x_i$ wird entweder auf dem KM oder mittels Auswertungen von $f$ PS einmal durchgeführt, um ein potentielles lokales Optimum zu bestimmen. Der daraus resultierende Punkt $x_{ps}$ wird ausgewertet und dem KM hinzugefügt. Darüber hinaus wird auch dieser Punkt als untersucht markiert. In einzelnen Fällen gilt $x_{ps}$ gleich $x_s$, so dass kein besserer Punkt gefunden wurde. In diesen Fällen wird $x_{ps}$ nicht ausgewertet und nicht dem KM hinzugefügt, da der Punkt bereits bekannt ist.

Durch dieses Verfahren wird in den meisten Schritten dem KM ein neuer Punkt $x_{ps}$ hinzugefügt und so die Approximation verbessert. Darüber hinaus wird der zu untersuchende Bereich verkleinert, indem die Punkte innerhalb eines Radius $h_r$ um

die Punkte $x_s$ und $x_{ps}$ nicht weiter berücksichtigt werden. Die Suche konzentriert sich so vor allem auf unbekannte Bereiche. Durch dieses Verfahren werden jedoch vor allem zu Beginn untersuchte Bereiche ggf. nur schlecht untersucht, da das KM zu Beginn in einigen Fällen noch keine ausreichend gute Approximation der Funktion $f$ ist. Daher wird das Verfahren so erweitert, dass bereits untersuchte Punkte nur für eine bestimmte Anzahl von Iterationen markiert bleiben. Zu diesem Zweck erhält die Markierung einen Zeitstempel. Eine Markierung $m_i$ zu einem Punkt $x_i$ enthält den Zeitpunkt zu dem der Punkt zuletzt untersucht wurde. Zu Beginn wird für alle Punkte die Markierung auf $-\infty$ gesetzt. Wird ein Punkt $x_i$ markiert, so wird $m_i$ auf die Nummer der aktuellen Iteration gesetzt. Für spätere Überprüfungen gilt $x_i$ als markiert, wenn die Nummer der aktuellen Iteration maximal um $m_{max}$ größer als $m_i$ ist. Dabei ist $m_{max}$ das maximale Alter einer Markierung.

Der bereits erwähnte Radius $h_r$, der die Größe des markierten Bereichs festlegt, muss für die entsprechende Funktion passend gewählt werden. Wird ein zu großer Wert gewählt, wird ein zu großer Bereich markiert und so ggf. mehrere lokale Optima durch diesen verdeckt. Ist der Wert hingegen zu klein gewählt, kann der Bereich ggf. ein lokales Optimum nicht abdecken und so zu unnötigen weiteren Inspektionen dieses Bereichs führen.

Als weitere Modifikation wird eine Abwandlung des EI für HKO $\mathrm{EI}_{\mathrm{hko}}$ vorgeschlagen. Sie entspricht in Bereichen, die nicht als untersucht markiert sind, dem normalen EI. In bereits untersuchten Bereichen ist sie 0. Dadurch wird erreicht, dass bereits untersuchte Bereiche auch bei dem normalen Kriging-Teilschritt nicht weiter beachtet werden.

## 8.2. Versuche und Ergebnisse

In den im weiteren Verlauf präsentierten Versuchen wird die normale KMO mit verschiedenen Konfigurationen der HKO verglichen. Dabei werden folgende Konfigurationen untersucht. Der Wert für den Radius $h_r$ wird auf 0,1, 0,2 und 0,3 gesetzt. Die Inspektion durch PS wird jeweils einmal direkt mit der Funktion $f$ und einmal auf dem KM durchgeführt. Darüber hinaus wird einmal das normale EI und einmal $\mathrm{EI}_{\mathrm{hko}}$ genutzt. Jede Optimierung wurde solange durchgeführt, bis 120 Punkte dem KM hinzugefügt worden sind. Alle Experimente werden 100 mal wiederholt und für die drei Funktionen Sinus, Ackley und Schaffer durchgeführt.

In den Abbildungen 8.1, 8.2 und 8.3 sind die Ergebnisse der Experimente dargestellt. Dabei steht jedes Balkenpaar für eine Konfiguration, wobei der linke hellere Balken die durchschnittliche Distanz zum Optimum mit 90% Konfidenzinterval und der rechte dunklere Balken die durchschnittliche Anzahl an Auswertungen angibt. Die Reihenfolge der Konfigurationen in Tabelle 8.1 entspricht der Reihenfolge der Konfigurationen in den Abbildungen.

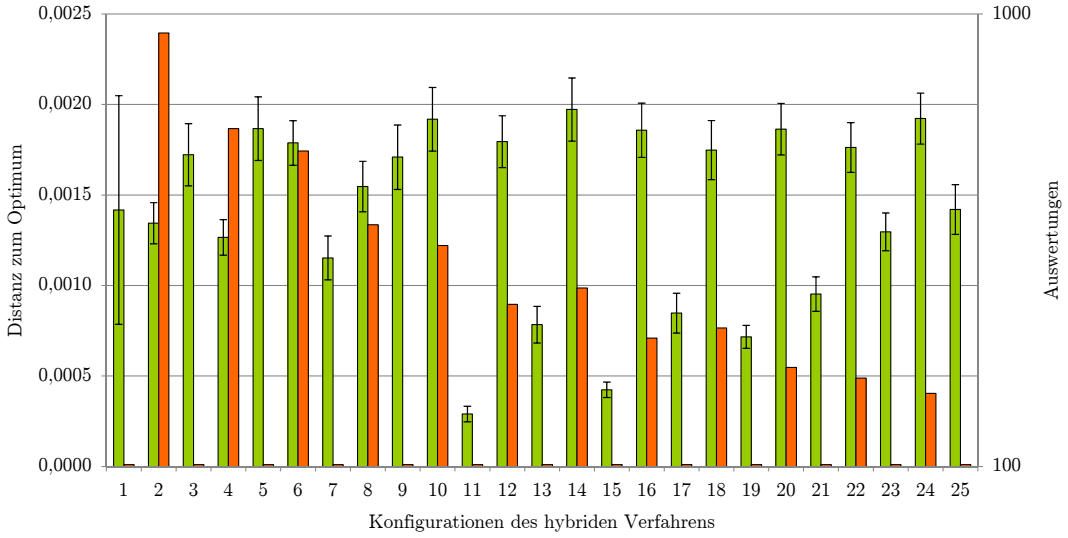



Abbildung 8.1.: Durchschnittliche Distanz zum Optimum (mit 90%-Konfidenzintervall) (grün) und Anzahl Auswertung für die normale KMO und HK (orange) in verschiedenen Konfigurationen (siehe Tabelle 8.1) am Beispiel der Sinus-Funktion.

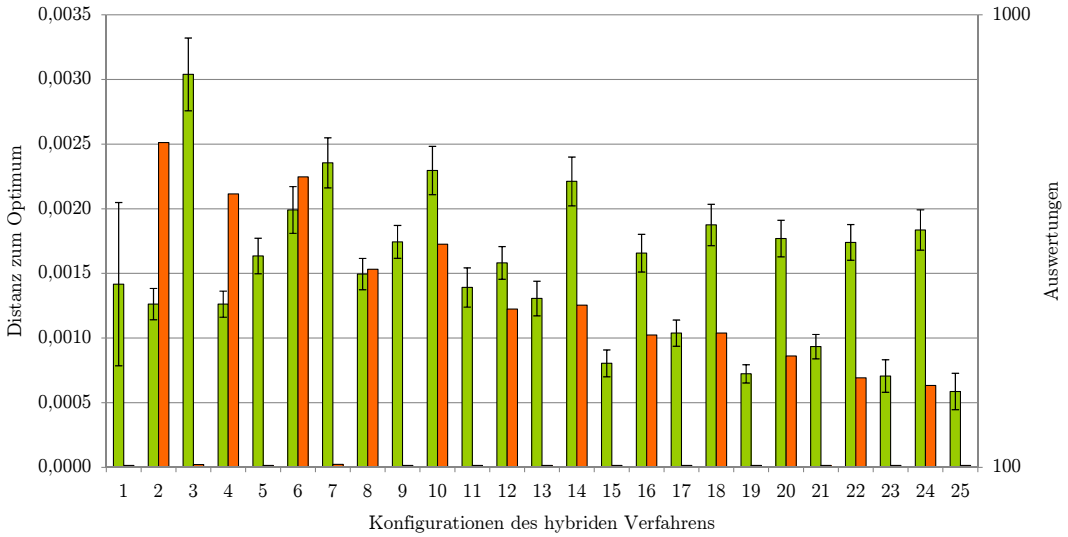



Abbildung 8.2.: Durchschnittliche Distanz zum Optimum (mit 90%-Konfidenzintervall) (grün) und Anzahl Auswertung für die normale KMO und HK (orange) in verschiedenen Konfigurationen (siehe Tabelle 8.1) am Beispiel der Ackley-Funktion.

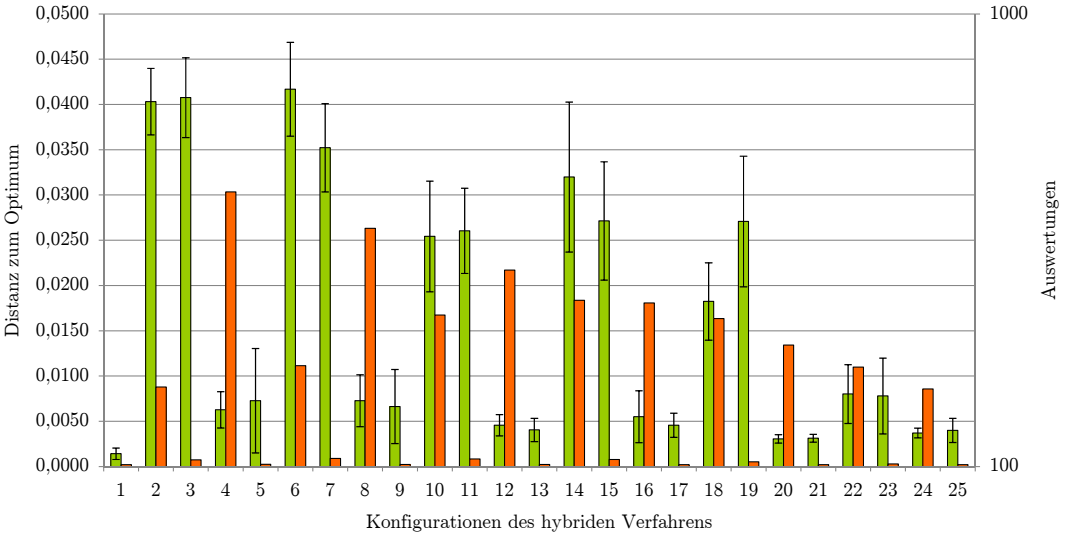



Abbildung 8.3.: Durchschnittliche Distanz zum Optimum (mit 90%-Konfidenzintervall) (grün) und Anzahl Auswertung für die normale KMO und HK (orange) in verschiedenen Konfigurationen (siehe Tabelle 8.1) am Beispiel der Schaffer-Funktion.

Aus den Abbildungen 8.1, 8.2 und 8.3 wird deutlich, dass nicht alle vorgetragenen Überlegungen in der Praxis Vorteile mit sich bringen. Überraschend ist besonders, dass die Versuche mit einer Inspektion auf dem realen Metamodell schlechtere Ergebnisse liefern, als die Versuche mit Inspektionen auf dem KM. Die Ursache liegt darin, dass bei einer Inspektion auf dem KM in jedem Fall zusätzliche Informationen gewonnen werden. Entweder wird das lokale Optimum im KM bestätigt oder der Wert wird entsprechend korrigiert und das KM kann daraufhin angepasst werden. Werden die Auswertungen hingegen auf dem realen Metamodell durchgeführt, können fehlerhafte Approximationen im Metamodell unentdeckt bleiben.

Ein uneinheitlicheres Bild entsteht beim Vergleich der beiden EI. Während bei der Sinus-Funktion $\mathrm{EI}_{\mathrm{hko}}$ zu besseren Ergebnissen führt, liefert bei der Ackley- und der Schaffer-Funktion das normale EI zu den besseren Ergebnissen. Da es je nach Metamodell teils deutliche Unterschiede zwischen den Ergebnissen gibt, können beide Ansätze hier nur als Alternativen betrachtet werden, von denen keine der anderen überlegen ist.

Betrachtet man zuletzt noch die Versuche mit unterschiedlichen Werten für $h_r$, so kann man feststellen, dass der Einfluss der gewählten Werte relativ gering ist. Bei der Sinus-Funktion werden die besten Ergebnisse für $h_r = 0,2$ und bei der Ackley- und der Schaffer-Funktion für $h_r = 0,3$ erzielt. Daher ist festzustellen, dass die Größe von

$h_r$ zwar einen Einfluss auf das Ergebnis der Optimierung hat, bei sinnvoll gewählten Werten aber keine signifikanten Unterschiede zu erwarten sind.

Zuletzt müssen noch KMO und HKO verglichen werden. In den Abbildungen ist zu erkennen, dass für jede Funktion mindestens eine Konfiguration von HKO existiert, die im Vergleich zu KMO überlegen oder zumindest gleichwertig ist. Am besten schneidet HKO dabei bei der Sinus- und der Ackley-Funktion ab. Lediglich bei der Schaffer-Funktion ist die KMO minimal besser. In allen Fällen ist aber die Konfiguration von HKO mit normalem EI, $h_r = 0,3$ mit Inspektionen auf dem KM zum normalen KMO konkurrenzfähig oder diesem überlegen. Abschließend lässt sich sagen, dass HKO eine Erweiterung der bestehenden KMO ist, die auf verschiedenen Funktionen bessere Ergebnisse als das bestehende KMO liefern kann.

# 9. Optimierung mit Kriging Metamodellen in der Simulation

Bei der Optimierung ereignisdiskreter Modelle kann der korrekte Wert der Funktion nicht genau bestimmt werden. Daher muss der erwartete Mittelwert und ggf. das zugehörige Konfidenzintervall genutzt werden, wobei ggf. Punkte mit sich überschneidenden Konfidenzintervallen nicht miteinander verglichen werden können. Da Kriging Metamodelle jedoch für deterministische Funktionen entwickelt wurden, muss dieser Zustand gesondert berücksichtigt werden.

In diesem Kapitel werden verschiedene Verfahren zum Umgang mit ereignisdiskreten Simulationsmodellen und KM vorgestellt. Zuerst werden einige Ansätze aus der Literatur beschrieben. Anschließend werden weitere Ansätze entwickelt. Zum einen werden klassische Ansätze des **Ranking and Selection** mit der KMO kombiniert. Zum anderen werden zwei Ansätze vorgestellt, die speziell auf KM zugeschnitten sind und deren Fehlerschätzung bzw. EI anpassen.

## 9.1. Aktuelle Entwicklungen

In den letzten Jahren haben KM Einzug in den Bereich der Simulation erhalten. Dabei muss beachtet werden, ob KM nur die stochastischen Eigenschaften der Messwerte besser wiedergeben oder ob das Metamodell für ein Optimierung verbessert werden soll. Im ersten Abschnitt werden Ansätze zur Verbesserung der Approximation dargestellt, in späteren Abschnitten folgen Ansätze zur Optimierung mittels KM. Dieser Abschnitt soll vor allem Alternativen zu den von mir vorgestellten Anpassungen vorstellen. Diese werden jedoch nicht genauer untersucht und beurteilt.

In [ANS08] wird ein KM für stochastische Eingabedaten vorgeschlagen, bei dem zu jedem Punkt $x_i$ $m_i$ Replikationen existieren. Analog zum normalen Kriging wird das Ergebnis der $j$-ten Replikation an einem Punkt $x$ wie folgt angenommen:

$$y_j(x) = \mu(x) + \varepsilon(x) + \varepsilon_j(x) \tag{9.1}$$

$\varepsilon_j(x)$ gibt dabei den Messfehler der $j$-ten Replikation wieder.

In der Beschreibung beschränken sich die Autoren auf das normale Kriging und nehmen daher $\mu(x) = \mu$ an. Der Mittelwert der Messungen eines Punkts $x$ ist definiert als

$$\bar{y}(x_i) = \frac{1}{m_i} \sum_{j=1}^{m_i} y_j(x_i) \ . \tag{9.2}$$

Der Vektor der Funktionswerte $y$ ändert sich damit zu $y = (\bar{y}(x_1), \ldots, \bar{y}(x_n))'$. Der Schätzer der normalen KM wird erweitert, so dass zusätzlich ein Teil den Messfehler der Replikationen berücksichtigt:

$$\hat{y}(x) = \mu + r'_M (R_M + R_\varepsilon)^{-1} (y - \mu 1_n) \tag{9.3}$$

Dabei deutet der Index $M$ einen Bezug zum KM und der Index $\varepsilon$ einen Bezug zum Messfehler der Replikationen an. $r_M$ ist eine Modifikation des Korrelationsvektors $r$ des Punkts $x$ zu den im KM vertretenen Punkten, $R_M$ ist eine modifizierte Version der Korrelationsmatrix $R$ und $R_\varepsilon$ ist eine neue Matrix, die den Einfluss der Messfehler an den gemessenen Punkten berücksichtigt.

Die Autoren haben anhand eines Beispiels gezeigt, dass diese Methode gute Approximationen mit einer passenden Fehlererwartung ermöglicht. Einen ähnlichen Ansatz verfolgt [YNN08], wobei hier ausschließlich die Korrelationsmatrix $R$ angepasst wird. Die Autoren schlagen zwei Varianten vor:

$$R'_1 = R + \eta_c, R'_2 = R + \eta_v \tag{9.4}$$

Dabei sind $\eta_c$ und $\eta_v$ $n \times n$-Diagonalmatrizen in der Form:

$$\eta_c = \begin{pmatrix} c_0/c_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & c_0/c_1 \end{pmatrix} \tag{9.5}$$

$$\eta_v = \begin{pmatrix} c_1^*/c_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & c_n^*/c_1 \end{pmatrix} \tag{9.6}$$

$c_0$ beschreibt die erwartete Störung der Messung und wird über den zu wählenden Korrekturfaktor $c_1$ relativiert. Für $\eta_v$ wird die Konstante $c_0$ durch die Varianz des Messfehlers an dem Punkt $x_i$ ersetzt. Diese Modifikation führt genauso wie die in [ANS08] beschriebenen Änderungen dazu, dass die durchschnittlichen Funktionswerte der bekannten Punkte nicht exakt im Metamodell wiedergegeben werden müssen.

$c_0$ und $c_1$ sind dabei frei wählbare Werte, wobei in [YNN08] Schranken angegeben werden, die Bereiche mit wahrscheinlich guten Werten festlegen.

In [vB05b] und [Kle07a] wird unter anderem ein neuer Schätzer für die Varianz in KM vorgeschlagen und dabei Bootstrapping für KM eingesetzt. Nach Kleijnen wird die Varianz im Modell unterschätzt, wenn von deterministischen Werten zu stochastischen Werten gewechselt wird (von $f(x)$ zu $1/n \sum_{i=1}^{n} (f(x) + \varepsilon)$). Er schlägt vor, die Varianz mit Cross-Validation und Jackknifing zu bestimmen. Dabei werden einzelne Punkte aus dem Metamodell entfernt und die Schätzung dieser Punkte mit den bekannten Funktionswerten verglichen. Der Jackknifingfehler an einem Punkt $x_j$ bzgl. eines Punkts $x_i$ berechnet sich dabei wie folgt:

$$J_{j,i} = n\hat{y}_j(x_i) - (n-1)\hat{y}_{j;-i}(x_i) \tag{9.7}$$

Dabei ist $\hat{y}_{j;-i}(x_i)$ der Schätzer an Punkt $x_i$ für das Metamodell, das alle Punkte $x_1, \ldots, x_n$ außer $x_i$ enthält. $c$ ist die Anzahl der Kandidatenpunkte. Die Varianz berechnet sich dann wie folgt:

$$\hat{var}(J_j) = \frac{\sum_{i=1}^{n} J_{j,i} - \bar{J_j}^2}{n-1} \quad \text{mit} \quad \bar{J_j} = \frac{1}{n} \sum_{i=1}^{n} J_{j,i} \tag{9.8}$$

Eine detailliertere Beschreibung der Cross-Validation und des Jackknifings sowie weitere Ansätze finden sich darüber hinaus in der Dissertation von van Beers [vB05a].

Ein alternativer Ansatz zur Berücksichtigung von stochastisch bedingten Messfehlern wird mit dem „Bayesian fuzzy kriging“ in [BG00] vorgeschlagen. Dabei werden normale KM um einen bayesischen Teil, der Vorkenntnisse über das Modell einbringt, und einen Teil aus der Fuzzylogik zur Repräsentation der Ungenauigkeit der Messungen erweitert. Eine anwendungsorientierte Beschreibung von KM zur Anwendung auf stochastische Simulation findet sich darüber hinaus in [BKvBvN07].

Die Anpassung bzw. Erweiterung von KM auf stochastische Funktionen muss darüber hinaus in Verbindung mit der Optimierung betrachtet werden. So heben die in den vorherigen Abschnitten dargestellten Änderungen eine Eigenschaft von KM auf: Der erwartete Fehler im Metamodell ist an bekannten Punkten nicht mehr in allen Fällen 0. Da die meisten Optimierverfahren für KM auf das EI oder ein ähnliches Verfahren setzen, müssen jedoch die Auswirkungen dieser Änderung berücksichtigt werden.

In [HANZ06] wird der EGO-Algorithmus aus [JSW98] für stochastische Funktionen erweitert. Dieses Verfahren nennen die Autoren „Sequentielle Kriging Optimierung“ (SKO). Dabei wird als „exakter“ Funktionswert eines Punkts $x$ der Mittelwert der Messwerte an diesem Punkt angenommen. Die Modifikation betrifft im Wesentlichen das EI. Dieses wird wie folgt abgewandelt:

$$EI_{sko}(x) := E\left[max\left(\hat{y}(x_{sko}) - Y_x, 0\right)\right] \cdot \left(1 - \frac{\sigma}{\sqrt{s^2(x) + \sigma^2}}\right) \tag{9.9}$$

Der benötigte Erwartungswert $E\left[max\left(\hat{y}(x_{sko}) - Y_x, 0\right)\right]$ berechnet sich wie folgt:

$$\begin{aligned} E\left[max\left(\hat{y}(x_{sko}) - Y_x, 0\right)\right] &= \left(\hat{y}(x_{sko}) - Y_x\right) \Phi\left(\frac{\hat{y}(x_{sko}) - Y_x}{s(x)}\right) \\ &\quad + s(x)\phi\left(\frac{\hat{y}(x_{sko}) - Y_x}{s(x)}\right) \end{aligned} \tag{9.10}$$

Der Punkt $x_{sko}$ übernimmt in dieser Formel die Rolle des Punkts mit dem Funktionswert $f_{min}$ im EI der normalen KM. Es ist dementsprechend der Punkt mit dem niedrigsten erwarteten Funktionswert. Da der Funktionswert nicht exakt bekannt ist, wird der erwartete Fehler $s(x)$ mit der Konstanten $c$ in die Berechnung einbezogen. Der Punkt $x_{sko}$ berechnet sich daher als:

$$argmax_{x_1,\ldots,x_n}\left(u(x)\right) \quad \text{mit} \quad u(x) = -y(x) - c \cdot s(x) \tag{9.11}$$

Ist die Konstante $c$ ausreichend groß gewählt, so ist das Expected Improvement an allen bekannten Punkten 0. Größere Werte von $c$ führen jedoch dazu, dass sich das EI im gesamten Wertebereich 0 annähert. Eine Erweiterung dieser Methode „Multiple Fidelity Sequential Kriging Optimization“ stellt Huang in seiner Dissertation [Hua05] vor.

Die hier vorgestellten Ansätze zeigen, dass es eine Reihe von Möglichkeiten gibt, KM an stochastische Funktionen anzupassen. Sie zeigen aber auch, dass diese Möglichkeiten erst seit relativ kurzer Zeit untersucht werden. Dies hängt sicherlich mit dem nötigen Rechenaufwand der KM und der ständig steigenden Rechenleistung in Computern zusammen. Es ist dadurch aber auch klar, dass es in diesem Bereich noch viel Potential für Forschungsarbeiten gibt. Einige Möglichkeiten, die dieser Bereich bietet, werden im folgenden genauer betrachtet.

## 9.2. Dynamische Anpassung der Replikationen zur Verbesserung der Modellqualität

Bei den bisher vorgestellten Kriging Metamodellen ist es wichtig, ausreichend verlässliche Werte zur Berechnung der Metamodelle zu nutzen. Dies bedeutet insbesondere, dass sich die Konfidenzintervalle zweier nebeneinander liegender Punkte nicht überschneiden dürfen, um das Verhalten der Funktion angemessen approximieren zu können. Zu diesem Zweck muss zur Laufzeit entschieden werden, welche Punkte erneut ausgewertet werden müssen.

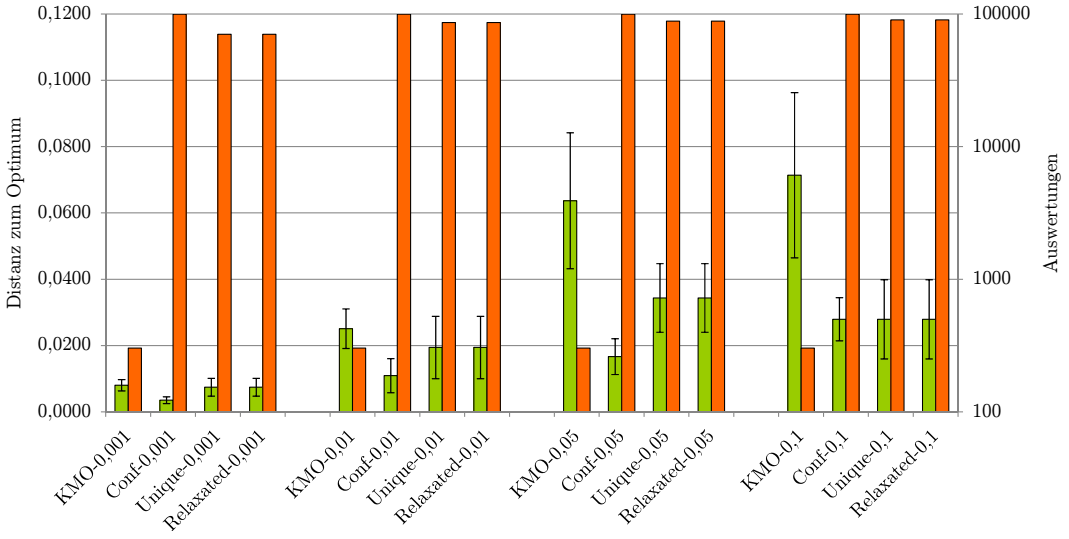



Abbildung 9.1.: Durchschnittliche Distanz zum Optimum und Anzahl Auswertung für unterschiedliche Strategien zur dynamischen Anpassung der Replikationen am Beispiel der Sinus-Funktion

In diesem Abschnitt werden drei Strategien vorgeschlagen, mit denen die Anzahl der Auswertungen dynamisch bestimmt wird. Bei der ersten Strategie „Conf“ werden alle Punkte solange ausgewertet, bis das Konfidenzintervall eine vorher zu definierende Größe $c$ nicht überschreitet. Die Strategie „Unique“ wertet alle Punkte solange aus, bis sich die Konfidenzintervalle aller Punkte nicht mehr überschneiden. Zu diesem Zweck werden alle Punkte paarweise verglichen. Überschneiden sich die Konfidenzintervalle wird für den Punkt mit bisher weniger Auswertungen eine weitere Auswertung durchgeführt. Die Strategie „Relaxated“ entspricht im Wesentlichen der Strategie „Unique“, jedoch wird hier eine Überschneidungszone $d^*$ akzeptiert.

In diesem Abschnitt werden die beschriebenen Strategien getestet und mit dem Verhalten der normalen KMO verglichen. Hierzu werden jeweils 100 Replikationen des KMO durchgeführt und der durchschnittliche Abstand zum Optimum sowie die durchschnittliche Anzahl an Auswertungen gemessen. Für diese Experimente wurde $d^* = 0,01$ gewählt. Die Experimente wurden für die Funktionen Sinus, Ackley und Schaffer durchgeführt, die jeweils mit einer normalverteilten Störung mit Mittelwert 0 versehen wurden. Die Varianz wurde auf $0,001$, $0,01$, $0,05$ und $0,1$ gesetzt.

In den Abbildungen 9.1, 9.2 und 9.3 sind die Ergebnisse dieser Experimente dargestellt. Jedes Balkenpaar steht dabei für eine getestete Konfiguration, wobei der linke, hellere Balken die durchschnittliche Distanz zum Optimum und der rechte, dunklere Balken die durchschnittliche Anzahl Auswertungen darstellt. Zur durchschnittlichen Distanz zum Optimum ist zusätzlich das 90%-Konfidenzintervall angegeben. In jeder Grafik existieren vier Gruppen von Balkenpaaren. Innerhalb einer Gruppe wurde je-

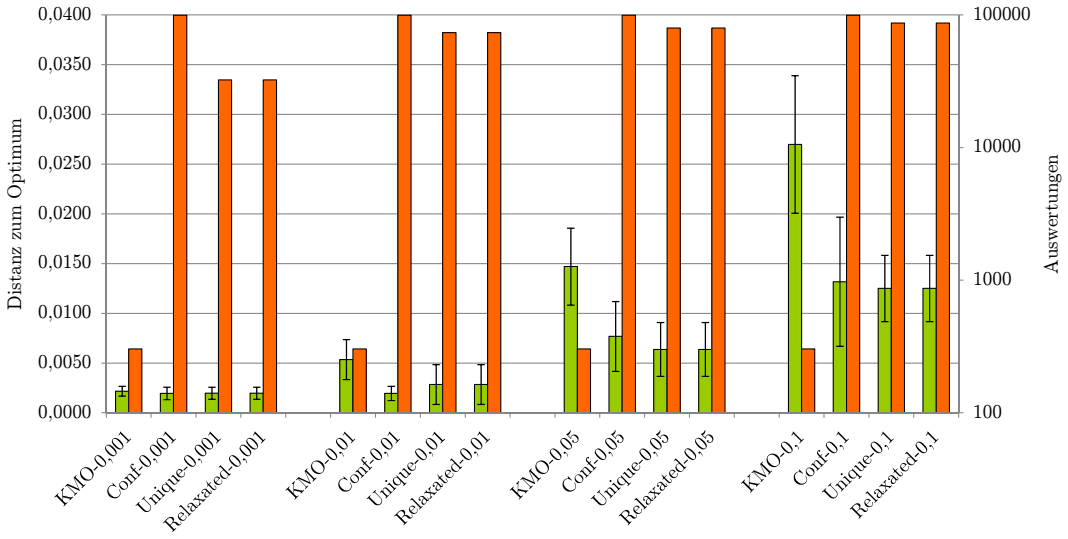



Abbildung 9.2.: Durchschnittliche Distanz zum Optimum und Anzahl Auswertung für unterschiedliche Strategien zur dynamischen Anpassung der Replikationen am Beispiel der Ackley-Funktion

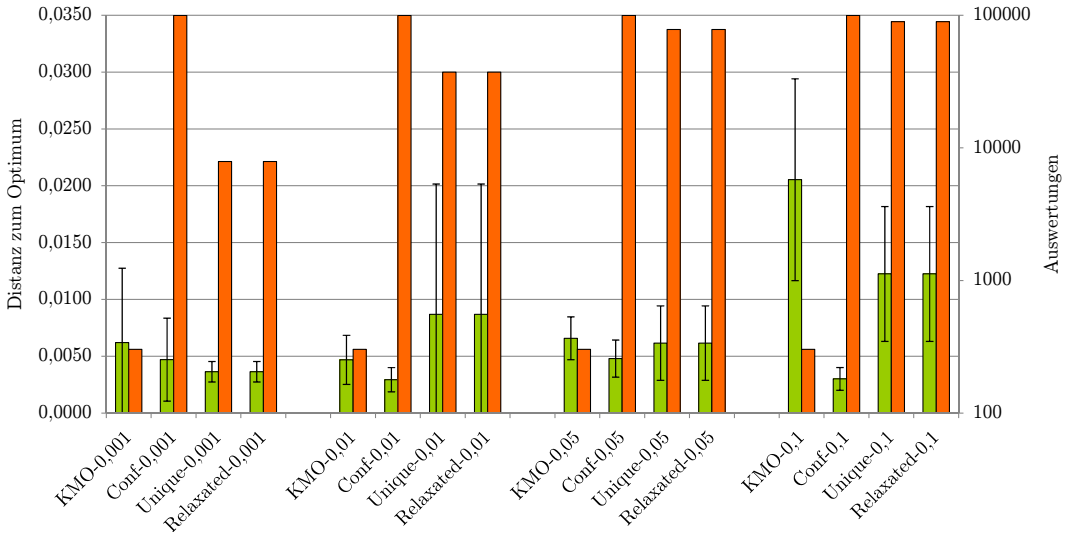



Abbildung 9.3.: Durchschnittliche Distanz zum Optimum und Anzahl Auswertung für unterschiedliche Strategien zur dynamischen Anpassung der Replikationen am Beispiel der Schaffer-Funktion

weils die Varianz für die Störung konstant gewählt. Diese steigt von links nach rechts mit jeder Gruppe von $0,001$ bis $0,1$. Innerhalb jeder Gruppe sind die Balkenpaare einer Strategie zugeordnet. Das jeweils linke Balkenpaar entspricht der normalen KMO. Von links nach rechts folgen die Strategien: Conf, Unique und Relaxated.

Auf den ersten Blick werden zwei Dinge deutlich: Zum einen liefern die vorgestellten Strategien deutlich niedrigere Distanzen zum Optimum als KMO, zum anderen ist die Anzahl Auswertungen deutlich höher. Die niedrigsten Distanzen werden in den meisten Fällen mit der Strategie Conf erzielt. Festzuhalten bleibt aber auch, dass diese Verfahren nicht gezielt auf die Optimierung mit KMO zugeschnitten sind, sondern die gesamte Qualität des Metamodells verbessern. Zur Optimierung ereignisdiskreter Modelle mittels KM sind andere Verfahren voraussichtlich besser geeignet.

## 9.3. Optimierung mit Kriging Metamodellen und „Ranking and Selection“

Bei der Optimierung mittels normaler KM ist es wichtig, ausreichend verlässliche Werte zur Berechnung der Metamodelle zu nutzen, da der beste Punkt am Ende des Algorithmus als Ergebnis ausgegeben wird. Verschiedene „Ranking and Selection“ Verfahren (RnS) ermöglichen auch bei stochastischen Funktionen mit einer hohen Wahrscheinlichkeit den richtigen Punkt aus einer Menge von Kandidaten zu wählen. Ursprünglich wurden RnS als eigenständiges Verfahren zur Auswahl von Kandidaten aus einer Menge von Konfigurationen eingesetzt. Mittlerweile werden sie auch erfolgreich in Kombination mit anderen Verfahren eingesetzt, so sind sie z.B. Evolutionären Algorithmen [BT05] kombiniert worden. Dort dienen sie zur Auswahl der besten Punkte aus einer Menge von Kandidaten (siehe Kapitel 5.7). Sie wurden darüber hinaus bereits erfolgreich in einer Variante von PS, dem sogenannten „Generalized Pattern Search“, in [Sri04] eingesetzt. In den folgenden Abschnitten wird zuerst eine Übersicht über die integrierten RnS gegeben, anschließend wird die Integration in die Optimierung mittels KM beschrieben und abschließend werden einige Versuche und Ergebnisse präsentiert.

### 9.3.1. Ranking and Selection

Verfahren zum RnS stellen ein weites Forschungsfeld dar. Die Arbeiten in diesem Feld sind nicht abgeschlossen, obwohl bereits einige RnS existieren. Insbesondere für verschiedene Sonderfälle (siehe z.B. [BM09]) besteht weiterhin Forschungsbedarf. Für einen detaillierten Einstieg in das Thema wird auf [BSG95] verwiesen. Dieser Abschnitt beschränkt sich auf die grundlegenden Aspekte und die in dieser Arbeit betrachteten Verfahren. Die hier folgende Beschreibung orientiert sich an [BT05].

RnS dienen dazu, aus einer Menge $N$ mit $n$ Kandidaten die besten $m$ Punkte auszuwählen. Da der Funktionswert bei stochastischen Funktionen nicht exakt bestimmt werden kann, kann nicht garantiert werden, die besten $m$ Punkte auszuwählen. Die im Weiteren betrachteten RnS beschränken sich daher meist auf folgende Aussage: Es werden aus einer Menge $N$ mit $n$ Kandidaten mit einer Wahrscheinlichkeit von $P^*$ die besten $m$ Punkte ausgewählt, wenn nach Sortierung der Kandidaten nach den Erwartungswerten die Differenz der Punkten mit Index $m$ und $m+1$ der Erwartungswerte nicht kleiner als $d^*$ ist. Dabei ist $d^*$ der Parameter der Unentscheidbarkeitszone (engl. indifference-zone) und legt fest, dass zwei Punkte, deren reale Funktionswerte nicht weiter als $d^*$ auseinander liegen, nicht weiter unterschieden werden. Einige dieser Verfahren sind ausschließlich für $m = 1$ ausgelegt. Bei allen Verfahren muss darüber hinaus beachtet werden, dass die Wahrscheinlichkeit für den Worst-Case berechnet wurde, um die Wahrscheinlichkeit $P^*$ zu garantieren. Dieser tritt nur für $y(x_1) + d^* = y(x_2) = \ldots = y(x_n)$ ein. Die Verfahren sind daher in der Regel eher konservativ.

In [Rin78] wird ein zweistufiges RnS vorgeschlagen, dass mittlerweile die Basis für eine Reihe weiterer RnS darstellt. In diesem Verfahren wird davon ausgegangen, dass in der ersten Stufe für alle Punkte $n_0$ Auswertungen durchgeführt werden. Die Mittelwerte dieser $n_0$ Auswertungen eines Punkts $x$ werden als $\bar{y}_{n_0}(x)$ und die Varianz der Auswertungen als $s^2_{n_0}(x)$ bezeichnet und berechnen sich wie folgt:

$$\bar{y}_{n_0}(x) = \frac{1}{n_0} \sum_{j=1}^{n_0} y_j(x) \tag{9.12}$$

$$s^2_{n_0}(x) = \frac{1}{n_0 - 1} \sum_{j=1}^{n_0} \left( y_j(x) - \bar{y}_{n_0}(x) \right)^2 \tag{9.13}$$

$y_j(x)$ ist dabei das Ergebnis der $j$-ten Auswertung des Punkts $x$. Aus diesen Werten wird die Anzahl der benötigten Auswertungen $N(x)$ für die Stufen 1 und 2 berechnet:

$$N(x) = max\left( n_0; \left\lceil \left( \frac{h}{d^*} \right)^2 s^2_{n0}(x) \right\rceil \right) \tag{9.14}$$

Dabei ist $h = h(n, P^*, n_0)$ eine Konstante, die Rinotts Integral (siehe [BSG95]) erfüllt. Mit Ergebnissen der Auswertungen aus der ersten Stufe und der zweiten Stufe wird das arithmetische Mittel berechnet und der Punkt $x$ mit dem kleinsten Mittelwert als bester Punkt ausgewählt. Dieser Punkt ist unter den oben angegebenen Voraussetzungen mit einer Wahrscheinlichkeit von $P^*$ der beste Punkt.

Aufbauend auf diesem Verfahren wurde von Chen und Kelten das Verfahren der „Enhanced Two-Stage Selection“[CK00] (ETSS) entwickelt, bei dem die Konstante $h$ durch $h_i$ ersetzt wird:

1: Setze $\mathcal{D} = \{x_1, \ldots, x_n\}$, $\bar{y}^* = \infty$ und $x^* = NULL$.
2: **while** $\mathcal{D} \neq \emptyset$ **do**
3: Wähle ein $x_i \in \mathcal{D}$ und setze $\mathcal{D} = \mathcal{D} \setminus \{x_i\}$.
4: Führe für $x_i$ $n_0$ Auswertungen durch.
5: $N_i = max\left\{n_0; \lceil h^2 s^2_{n_0}(x_i)/(d^*)^2 \rceil\right\}$;
6: **for** $r = n_0, \ldots, N_i$ **do**
7: Setze $W_{i,r} = max\left\{0; \frac{d^*}{2r}\left(h^2 s^2_{n_0}/(d^*)^2 - r\right)\right\}$.
8: **if** $\bar{y}_r(x_i) - W_{i,r} > \bar{y}^*$ **then**
9: **break**
10: **else**
11: **if** $\bar{y}_r(x_i) + W_{i,r} < \bar{y}^*$ **then**
12: Erhöhe die Anzahl der Auswertung von $x_i$ auf $N_i$.
13: **if** $\bar{y}_{N_i}(x_i) < \bar{y}^*$ **then**
14: Setze $x^* = x_i$ und $\bar{y}^* = \bar{y}_{N_i}(x_i)$.
15: **end if**
16: **break**
17: **end if**
18: **end if**
19: **if** $r < N_i$ **then**
20: Führe eine weitere Auswertung an der Stelle $x_i$ durch.
21: **end if**
22: **end for**
23: **end while**
24: Gib $x^*$ als besten Punkt aus.

**Algorithmus 9.1:** Pseudocode von QOG

$$h_i = \frac{hd^*}{max\left\{d^*; max_{j=1,\ldots,n}\left\{\bar{y}_{n_0}(x_i)\right\} - \bar{y}_{n_0}(x_i)\right\}} \tag{9.15}$$

Durch diese Ersetzung kann die Anzahl der Auswertungen deutlich reduziert werden, wodurch das Verfahren beschleunigt wird. Da $h$ jedoch durch $h_i$ ersetzt wurde, wird die garantierte Wahrscheinlichkeit von Rinotts Verfahren aufgehoben.

Ein anderes Verfahren BNK (nach den Autoren Boesel, Nelson und Kim [BNK05]) hebt die Einschränkung der gleichen Anzahl an Auswertungen je Punkt in der ersten Stufe des RnS auf. Jeder Punkt darf zu Beginn $n_{0i}$ Auswertungen enthalten. Darüber hinaus ersetzen sie die Konstante $h$ durch $h(2, (P^*)^{1/(n-1)}, min_{i=1,\ldots,n}\{n_{0i}\})$.

Mit einem Verfahren zur Vorauswahl (engl. subset pre-selection) können signifikant schlechtere Punkte bereits zu Beginn herausgefiltert werden. Beim „extended screen-to-the-best“-Verfahren [BNK05] wird eine Menge $H \subseteq N$ bestimmt, die die besten $|H|$ Punkte enthält, wobei $P\{x_k \in H | x_k - x_{k-1} \geq d^*\} \geq P^*$ gilt. Die Größe von $H$ kann jedoch nicht direkt beeinflusst werden und ist im schlechtesten Fall $n$. Durch

Kombination dieser Subset Selection mit dem bereits beschriebenen Verfahren BNK entsteht das „Combined Screening and Selection“ (CSS).

In [BT05] wurde eine iterative Methode „Iterative Subset Selection“ (ISS) vorgeschlagen. Diese Methode nutzt BNK zur Vorauswahl von Punkten. Damit die Größe von $H$ bis auf $m$ sinkt, werden iterativ weitere Auswertungen für alle Punkte in $H$ durchgeführt. Durch diese zusätzlich gewonnen Informationen ist es in der Regel möglich, $H$ ausreichend zu verkleinern. Genauso wie ETSS ist ISS eine Heuristik.

In [OK07] wird ein alternativer Algorithmus zur Bestimmung des besten Punkts „Quick Optimization through Guessing“ (QOG) vorgeschlagen. Bei diesem Verfahren werden iterativ einzelne Punkte untersucht und entweder als bisher bester Punkt identifiziert oder verworfen (siehe Algorithmus 9.1). Für die Berechnung einiger Werte wird eine Konstante $h_q$ benötigt, die dem minimalen $h$ entspricht, das folgende Formel erfüllt:

$$E\left[\left(1+exp\left(\sqrt{\frac{h^2Q_1}{n_0-1}\left(1+\frac{Q_1}{Q_2}\right)}Z+\frac{h^2Q_1}{n_0-1}\right)\right)^{-1}\right]\leq\frac{\alpha}{n-1} \tag{9.16}$$

$Z$ ist dabei eine standardnormalverteilte Zufallsvariable und $Q_1$ und $Q_2$ sind $\chi^2$ verteilte Zufallsvariablen mit $n_0-1$ Freiheitsgraden. Durch die Wahl von $h_q$ entsprechend dieser Formel wird sichergestellt, dass die Wahrscheinlichkeit, den besten von $n$ Punkten zu wählen, $\frac{\alpha}{n-1}$ beträgt, sofern der Abstand des besten und zweitbesten Punkts mindestens $d^*$ beträgt. Weitere Informationen zur Berechnung von $h_q$ finden sich in Anhang A.

Beim Vergleich zweier Punkte wird bei QOG auf einen maximalen Abstand $W_{i,r}$ zurückgegriffen, der eine erweiterte Unsicherheit durch noch fehlende Messungen berücksichtigt:

$$W_{i,r}=max\left\{0;\frac{d^*}{2r}\left(h_q^2s_{n_0}^2/(d^*)^2-r\right)\right\} \tag{9.17}$$

Ist ein untersuchter Punkt $x_i$ um diesen Wert schlechter als der beste bisher bekannte Punkt $x^*$, so wird $x_i$ verworfen. Ist $x_i$ um diesen Abstand besser, so wird $x_i$ als neuer bester Punkt akzeptiert. Der Wert von $W_{i,r}$ sinkt mit steigendem $r$, wobei $W_{i,r}=0$ für $r=N_i$ gilt.

### 9.3.2. Integration von „Ranking and Selection“ Verfahren in die Optimierung mit Kriging Metamodellen

Die hier beschriebene Integration von RnS in KMO soll sicherstellen, dass man zu bestimmten Zeitpunkten die besten $m$ Punkte von den schlechteren $n-m$ Punkten

unterscheiden kann. Es werden also keine Punkte aus dem KM entfernt. Es soll dadurch jedoch sichergestellt werden, dass zu diesen Zeitpunkten das KM die Bereiche in der Nähe der $m$ besten Punkte ausreichend gut abbildet. Dabei sind verschiedene Aspekte zu beachten. Die Durchführung von RnS bei jeder Iteration der KMO führt gerade zu Beginn des Verfahrens zu einer stark erhöhten Anzahl an Auswertungen sowie zu einer Erhöhung der Rechenzeit. Darüber hinaus können vor allem zu Beginn des Verfahrens zwei Situationen eintreten, bei denen der Einsatz von RnS nicht sinnvoll ist. Zum Einen können zwischen den durchschnittlichen Funktionswerten der einzelnen Punkte große Unterschiede bestehen, so dass ein RnS noch nicht benötigt wird. Zum Anderen können die bisherigen Werte besonders schlecht sein, so dass ein Aussortieren dieser Punkte zu einem späteren Zeitpunkt, zu dem deutlich bessere Punkte bekannt sind, einfacher wird. Gegen Ende des Verfahrens ist es jedoch wichtig, nah beieinander liegende Punkte unterscheiden zu können. Aus diesem Grund ist es sinnvoll, den Einsatz von RnS in der KMO konfigurierbar zu gestalten. Aus diesem Grund kann die Anzahl der Iterationen $i_{rns}$, die zwischen zwei Durchführungen des RnS vergehen, festgelegt werden. Über sie kann zu einem gewissen Grad sowohl die Anzahl der benötigten Auswertungen als auch die benötigte Rechenzeit gesteuert werden.

In der vorliegenden Implementierung kann neben den bereits erklärten Parametern $d^*$ und $P^*$ auch der Startwert für die Anzahl $m$ der voraussichtlich besten Punkte festgelegt werden. Im Verlauf wird $m$ an die aktuelle Situation angepasst. Der Wert von $m$ in der $i$-ten Iteration wird als $m_i$ bezeichnet. Zur Anpassung werden die Werte für die Iteration mit der ersten Anpassung $i_{min}$ und für die Iteration mit der letzten Anpassung $i_{max}$ benötigt. $m_i$ berechnet sich dann wie folgt:

$$m_i = \begin{cases} m & ; i \leq i_{min} \\ 1 & ; i \geq i_{max} \\ \left\lceil m * \left(1 - \frac{i - i_{min}}{i_{max} - i_{min}}\right) \right\rceil & ; \text{sonst} \end{cases} \tag{9.18}$$

Gegen Ende der Optimierung wird so mittels RnS nur noch der beste Punkte bestimmt, während zu Beginn mehrere Punkte gewählt werden können. In einer Erweiterung dieses Verfahrens soll neben der Bestimmung der besten $m$ Punkte die Unterscheidbarkeit dieser Punkte erhöht werden. Zu diesem Zweck werden die besten $m$ Punkte wiederholt mittels RnS untersucht, wobei in jedem Schritt ein weiterer Punkt ausgeschlossen wird. Dieses Vorgehen wird solange wiederholt, bis nur noch ein Punkt vorhanden ist. Bei diesen Verfahren ist zu beachten, dass die Punkte nicht aus dem KM entfernt werden. Das KM profitiert dabei ausschließlich durch die verbesserte Unterscheidbarkeit der Punkte.

Darüber hinaus wird eine Erweiterung der KMO mit RnS auf der Basis einer adaptiven Bestimmung von $d^*$ vorgeschlagen. Diese Überlegung basiert auf den Ergebnissen aus [KMT06]. Dort wurde festgestellt, dass die benötigte Genauigkeit der Ergebnisse im Laufe einer Optimierung ansteigt. Sie kann zu Beginn vergleichsweise niedrig

ausfallen, sollte gegen Ende der Optimierung jedoch ansteigen, um eine hohe Ergebnisqualität zu sichern. Dazu wurde das bisher beschriebene Verfahren so erweitert, dass es $d^*$ abhängig von den Funktionswerten der bisher bekannten Punkte anpasst, wobei die beiden besten Punkte $x_1$ und $x_2$ die Basis für diese Berechnung sind. Für ein initiales $d_0^*$, ein Gewicht $g$ sowie einen Modifikator $d_m^*$ ergibt sich $d^*$ als:

$$d^* = (1 - w) \cdot d_0^* + w \cdot d_m^* \cdot |y(x_1) - y(x_2)| \tag{9.19}$$

Das Gewicht $w$ legt dabei fest, wie stark der Einfluss des adaptiven Ansatzes ist; $w = 0$ entspricht den nicht adaptiven Verfahren. Der Modifikator $d_m^*$ legt die grundsätzliche Tendenz fest. Ist $d_m^*$ größer als 1 werden größere Werte für $d^*$ gewählt. Eine stärkere Absicherung gegen zu große $d^*$ ist jedoch für $d_m^*$ kleiner als 1 gegeben.

### 9.3.3. Versuche und Ergebnisse

In den folgenden Abschnitten werden die bisher vorgestellten Anpassungen in unterschiedlichen Experimenten verglichen und so die Leistungsfähigkeit empirisch untersucht. Dabei wird zuerst der Einfluss von RnS auf die Optimierung betrachtet. In der zweiten Experimentserie wird mit RnS eine detaillierte Untersuchung der Punkte durchgeführt. In der dritten Experimentserie wird $d^*$ adaptiv gewählt.

### 9.3.4. Kriging-Metamodell basierte Optimierung mit Ranking and Selection

In diesem Abschnitt werden die bisher beschriebenen RnS-Verfahren getestet und mit dem Verhalten der normalen KMO verglichen. Hierzu werden jeweils 100 Replikationen der Algorithmen durchgeführt und der durchschnittliche Abstand zum Optimum sowie die durchschnittliche Anzahl an Auswertungen gemessen. Für diese Experimente wurden $d^* = 0,01$, $P^* = 0,95$ und $m = 10$ gewählt. Die Experimente wurden für die Funktionen Sinus, Ackley und Schaffer durchgeführt, die jeweils mit einer normalverteilten Störung mit Mittelwert 0 versehen wurden. Die Varianz wurde auf $0,001$, $0,01$, $0,05$ und $0,1$ gesetzt.

In den Abbildungen 9.4, 9.5 und 9.6 sind die Ergebnisse dieser Experimente dargestellt. Jedes Balkenpaar steht dabei für eine getestete Konfiguration, wobei der linke, hellere Balken die durchschnittliche Distanz zum Optimum und der rechte, dunklere Balken die durchschnittliche Anzahl Auswertungen darstellt. In jeder Grafik existieren vier Gruppen aus Balkenpaaren. Innerhalb einer Gruppe wurde jeweils eine Varianz für die Störung konstant gewählt. Diese steigt von links nach rechts mit jeder Gruppe von $0,001$ bis $0,1$. Innerhalb jeder Gruppe sind die Balkenpaare einem RnS zugeordnet und das jeweils linke Balkenpaar entspricht der normalen

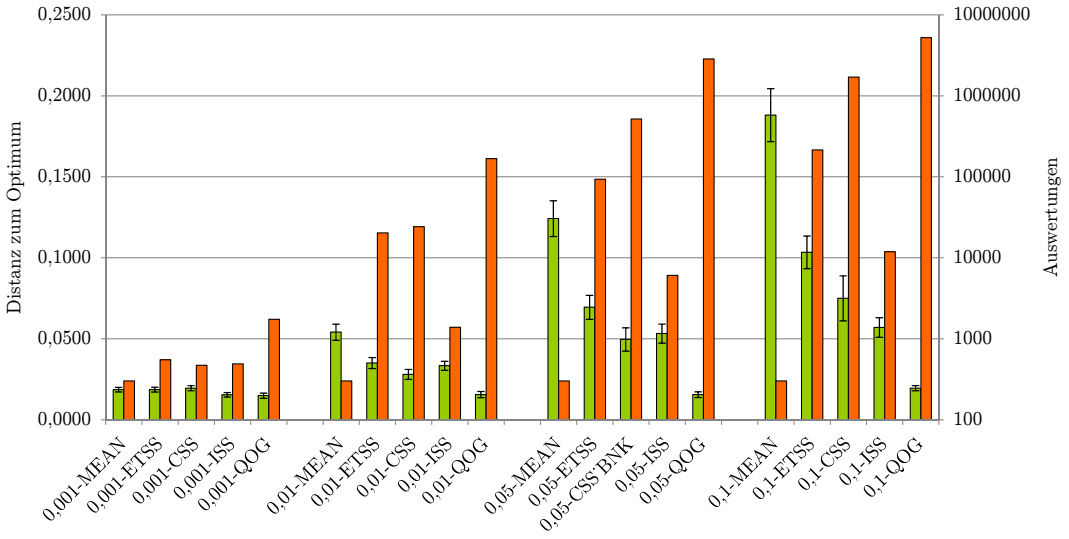



Abbildung 9.4.: Durchschnittliche Distanz zum Optimum und Anzahl Replikationen für unterschiedliche RnS-Algorithmen und Störungen am Beispiel der Sinus-Funktion

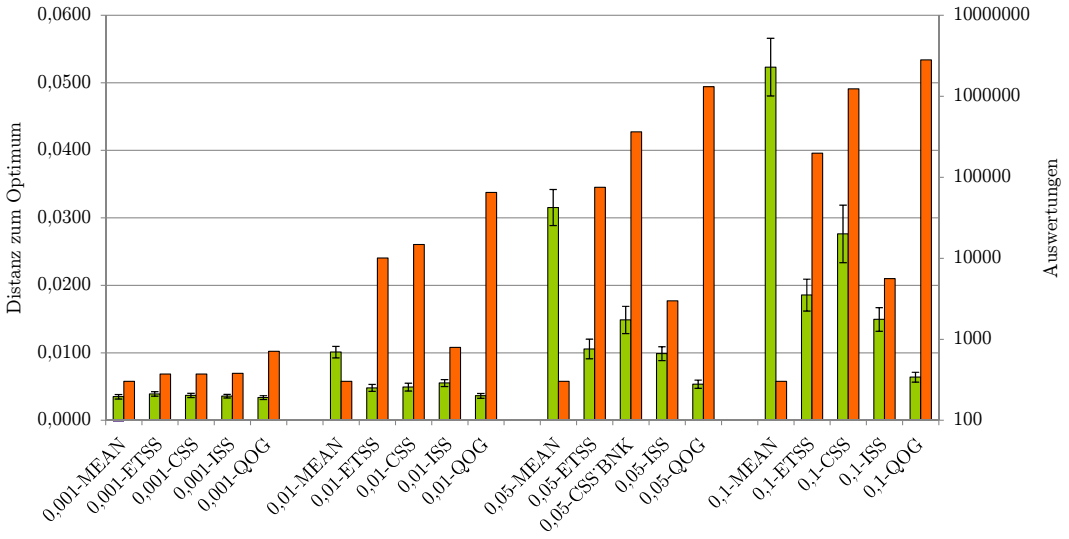



Abbildung 9.5.: Durchschnittliche Distanz zum Optimum und Anzahl Replikationen für unterschiedliche RnS-Algorithmen und Störungen am Beispiel der Ackley-Funktion

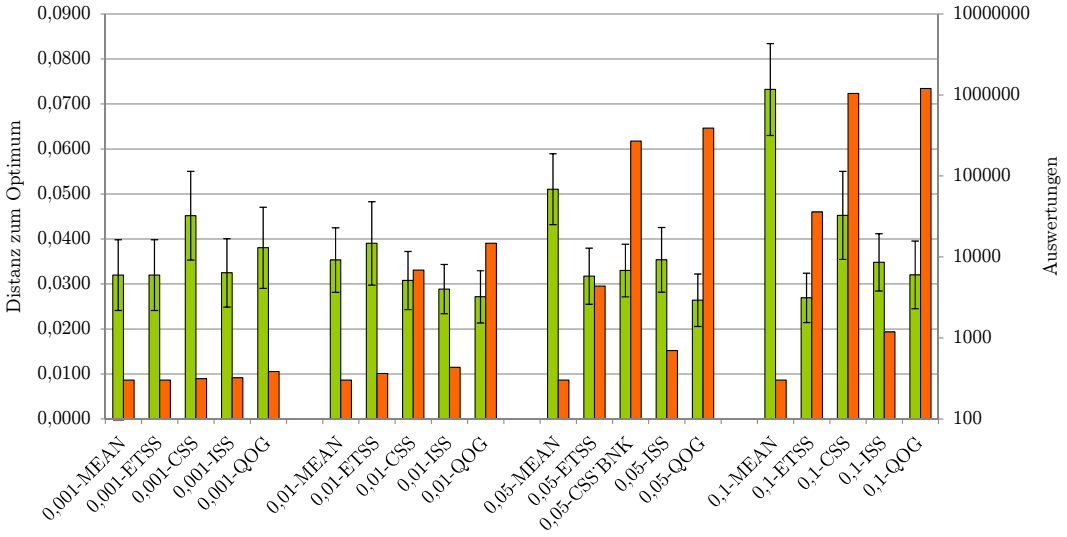



Abbildung 9.6.: Durchschnittliche Distanz zum Optimum und Anzahl Replikationen für unterschiedliche RnS-Algorithmen und Störungen am Beispiel der Schaffer-Funktion

KMO. Von links nach rechts sind die RnS wie folgt angeordnet: ETSS, CSS, ISS, QOG.

Auf den ersten Blick wird deutlich, dass die eingesetzten RnS deutlich mehr Auswertungen während der Optimierung als das normale KMO benötigen. Während jedoch die durchschnittliche Distanz zum Optimum für das normale KMO mit steigender Störung deutlich anwächst, kann die Distanz zum Optimum mit RnS deutlich geringer gehalten werden. Die Anzahl der Auswertungen wird dabei der Störung angepasst. Es überrascht nicht, dass konservative Verfahren wie CSS sehr viele Auswertungen benötigen. Das mit etwas abgeschwächten Bedingungen arbeitende ETSS kann bei weniger Auswertungen ähnliche Ergebnisse erzielen. Deutlich weniger Auswertungen als die ersten beiden Verfahren benötigt ISS und kann in den meisten Fällen eine geringere durchschnittliche Distanz zum Optimum vorweisen. Daher ist ISS in den betrachteten Fällen CSS und ETSS überlegen. Die durchschnittliche Distanz zum Optimum von ISS wird lediglich von QOG unterschritten, dies muss das Verfahren jedoch mit einer deutlich höheren Anzahl an Auswertungen erkaufen.

Anhand dieser Ergebnisse lässt sich feststellen, dass ISS ein gutes Verhältnis zwischen vielen Auswertungen und hoher Ergebnisqualität bietet. Wenn die Auswertungen wenig zeitintensiv sind, kann die Ergebnisqualität durch den Einsatz von QOG verbessert werden. Diese Ergebnisse sollen nun anhand einer Simulation der Modelle aus Kapitel 2.2.3 überprüft werden. Die Konfiguration bleibt weitestgehend unverändert. Da die Modelle bereits stochastische Elemente enthalten, wird den Ergebnissen keine zusätzliche Störung hinzugefügt.

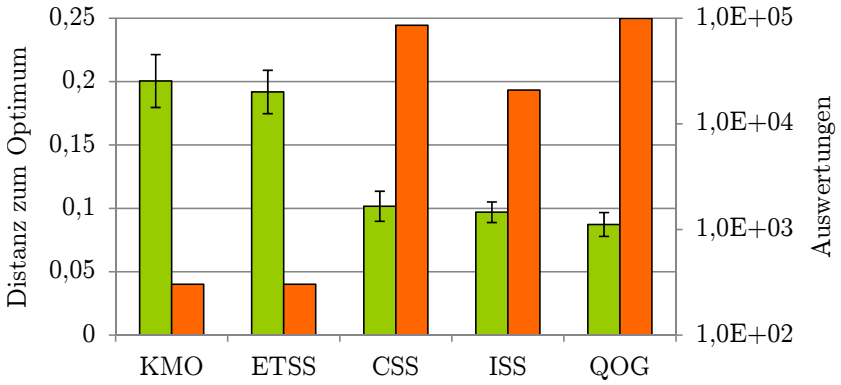



Abbildung 9.7.: Durchschnittliche Distanz zum Optimum und Anzahl Replikationen für unterschiedliche RnS-Algorithmen und Störungen am Beispiel einer Produktionslinie

Die Ergebnisse sind in Abbildungen 9.7 und 9.8 dargestellt. Sie ähneln den Ergebnissen aus dem vorherigen Abschnitt. Damit kann bestätigt werden, dass die Kombination von RnS und KMO gelungen ist und die RnS ISS und QOG zu den besten Ergebnissen führen, wobei abhängig von der zur Verfügung stehenden Zeit entweder ISS oder QOG gewählt werden sollte.

### Kriging-Metamodell basierte Optimierung mit intensivem „Ranking and Selection“

In diesem Abschnitt werden die bisher beschriebenen RnS-Verfahren getestet und mit dem Verhalten der normalen KMO verglichen. Als Erweiterung zu den Experimenten des letzten Abschnitts werden die RnS genutzt, um die Menge der besten Punkte schrittweise zu verkleinern. Zuerst werden die $m$ besten Punkte bestimmt. Anschließend wird jeweils ein weiterer Punkt ausgeschlossen, wodurch eine bessere Unterscheidung der besten Punkte gewährleistet werden soll. Hierzu werden jeweils 100 Replikationen der Algorithmen durchgeführt und der durchschnittliche Abstand zum Optimum sowie die durchschnittliche Anzahl an Auswertungen gemessen. Für diese Experimente wurden $d^* = 0,01$, $P^* = 0,95$ und $m = 10$ gewählt. Die Experimente wurden für die Funktionen Sinus, Ackley und Schaffer durchgeführt, die jeweils mit einer normalverteilten Störung mit Mittelwert 0 versehen wurden. Die Varianz wurde auf $0,001$, $0,01$, $0,05$ und $0,1$ gesetzt.

Die Ergebnisse sind in den Abbildungen 9.9, 9.10 und 9.11 dargestellt, wobei der Aufbau der Abbildungen identisch zu denen der vorherigen Abschnitte ist. Durch einen Vergleich mit den Abbildungen 9.4, 9.5 und 9.6 wird deutlich, dass die Unterschiede der Verfahren gering sind. Im Fall der Sinus- und Ackley-Funktion konnte

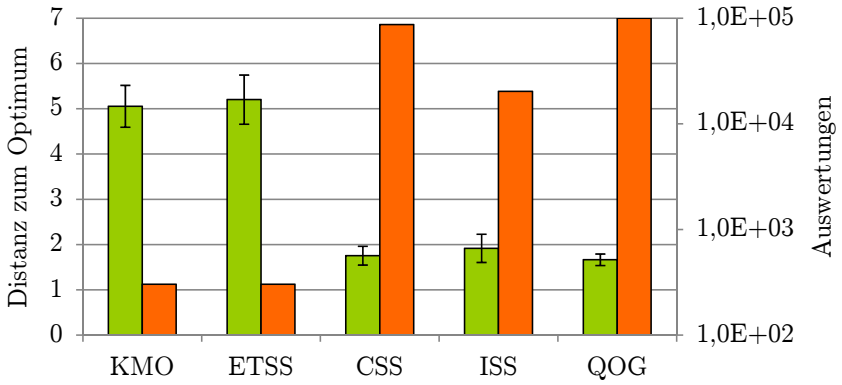



Abbildung 9.8.: Durchschnittliche Distanz zum Optimum und Anzahl Replikationen für unterschiedliche RnS-Algorithmen und Störungen am Beispiel eines Lagerhaltungssystems

auf Kosten zusätzlicher Auswertungen die Ergebnisqualität leicht gesteigert werden. Bei der Schaffer-Funktion ist die Güte der Ergebnisse beider Verfahren nahezu identisch, wobei auch hier das neue Verfahren zusätzliche Modellauswertungen benötigt. Daher kann nicht festgestellt werden, dass das neue Verfahren dem einfachen RnS überlegen ist. Wenn jedoch zusätzliche Auswertungen möglich sind, kann die Ergebnisqualität leicht verbessert werden.

### Kriging-Metamodell basierte Optimierung mit „Ranking and Selection“ und adaptivem $d^*$

In diesem Abschnitt werden die bisher beschriebenen RnS-Verfahren mit adaptivem $d^*$ getestet und mit dem Verhalten der normalen KMO verglichen. Hierzu werden jeweils 100 Replikationen der Algorithmen durchgeführt und der durchschnittliche Abstand zum Optimum sowie die durchschnittliche Anzahl an Auswertungen gemessen. Für diese Experimente wurden $d_0^* = 0,01$, $w = 0,25$, $d_m^* = 0,9$ $P^* = 0,95$ und $m = 10$ gewählt. Die Experimente wurden für die Funktionen Sinus, Ackley und Schaffer durchgeführt, die jeweils mit einer normalverteilten Störung mit Mittelwert 0 versehen wurden. Die Varianz wurde auf $0,001$, $0,01$, $0,05$ und $0,1$ gesetzt.

Die Ergebnisse der Experimente sind in den Abbildungen 9.12, 9.13 und 9.14 sowie in den Abbildungen 9.15, 9.16 und 9.17 dargestellt. Die Abbildungen entsprechen in dieser Reihenfolge den Abbildungen der Benchmarkfunktionen aus den vorherigen Abschnitten. Beim Vergleich dieser Ergebnisse kann festgestellt werden, dass sich die Ergebnisse nur minimal unterscheiden. Dies trifft sowohl auf die Ergebnisgüte als auch auf die Anzahl der benötigten Auswertungen zu. Die adaptiven Verfahren

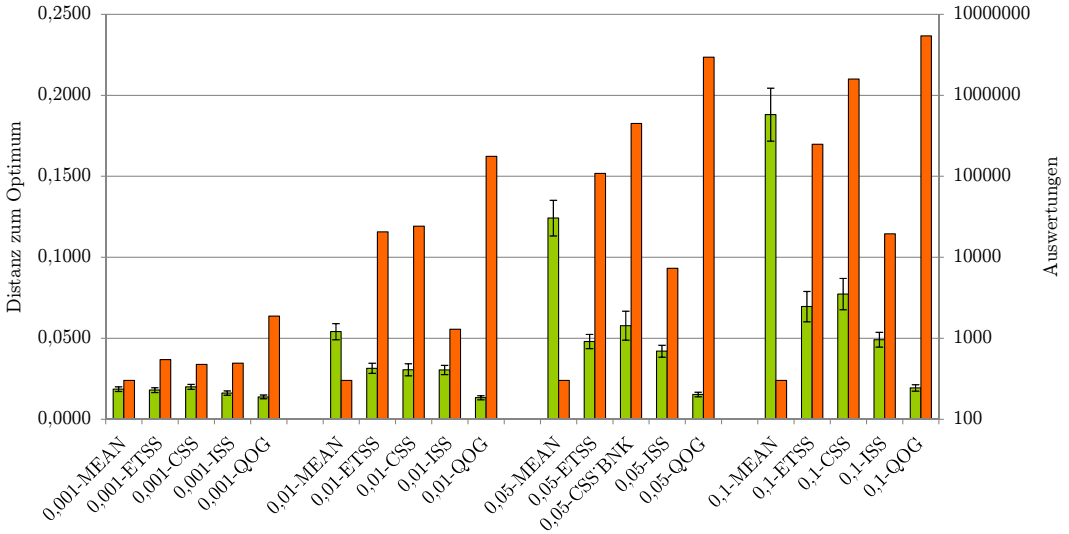



Abbildung 9.9.: Durchschnittliche Distanz zum Optimum und Anzahl Replikationen für unterschiedliche RnS-Algorithmen und Störungen am Beispiel der Sinus-Funktion mit intensiver Untersuchung

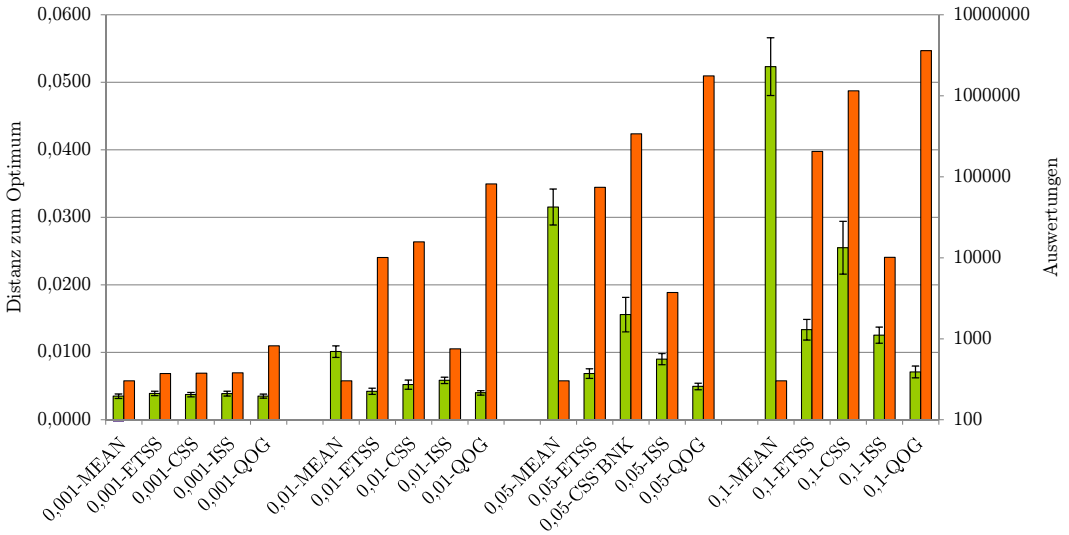



Abbildung 9.10.: Durchschnittliche Distanz zum Optimum und Anzahl Replikationen für unterschiedliche RnS-Algorithmen und Störungen am Beispiel der Ackley-Funktion mit intensiver Untersuchung

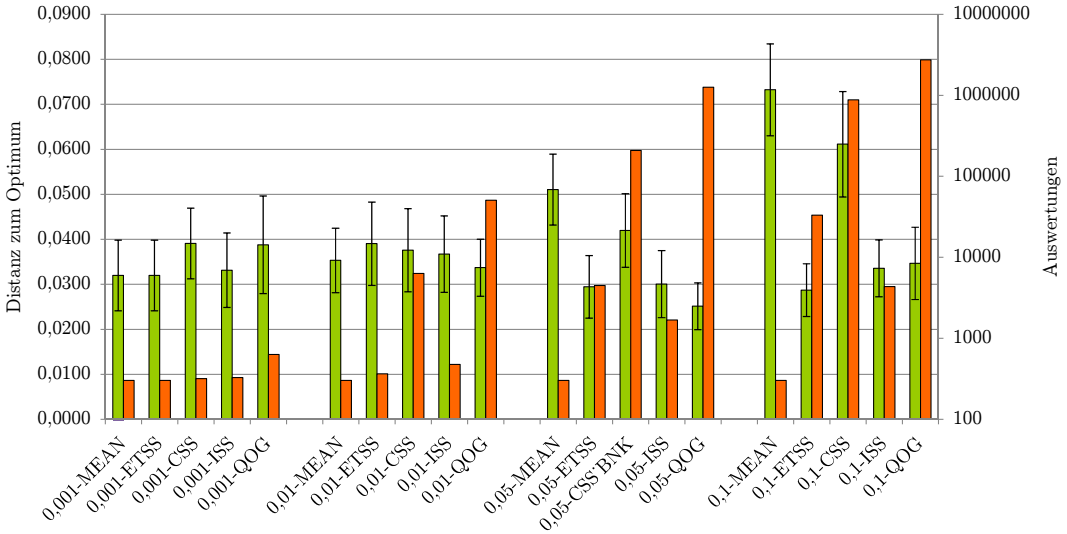



Abbildung 9.11.: Durchschnittliche Distanz zum Optimum und Anzahl Replikationen für unterschiedliche RnS-Algorithmen und Störungen am Beispiel der Schaffer-Funktion mit intensiver Untersuchung

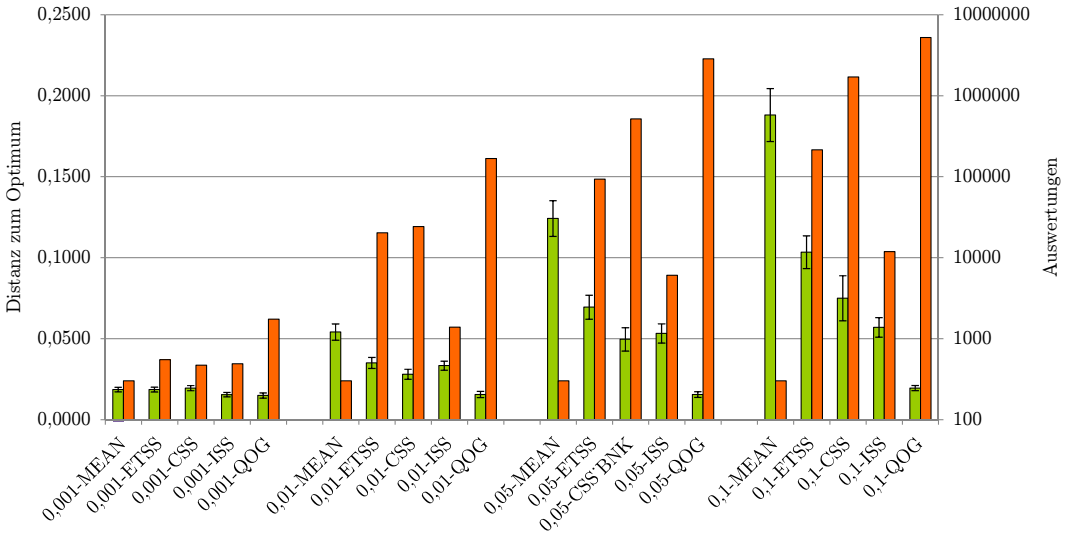



Abbildung 9.12.: Durchschnittliche Distanz zum Optimum und Anzahl Replikationen für unterschiedliche RnS-Algorithmen und Störungen am Beispiel der Sinus-Funktion und adaptivem $d^*$

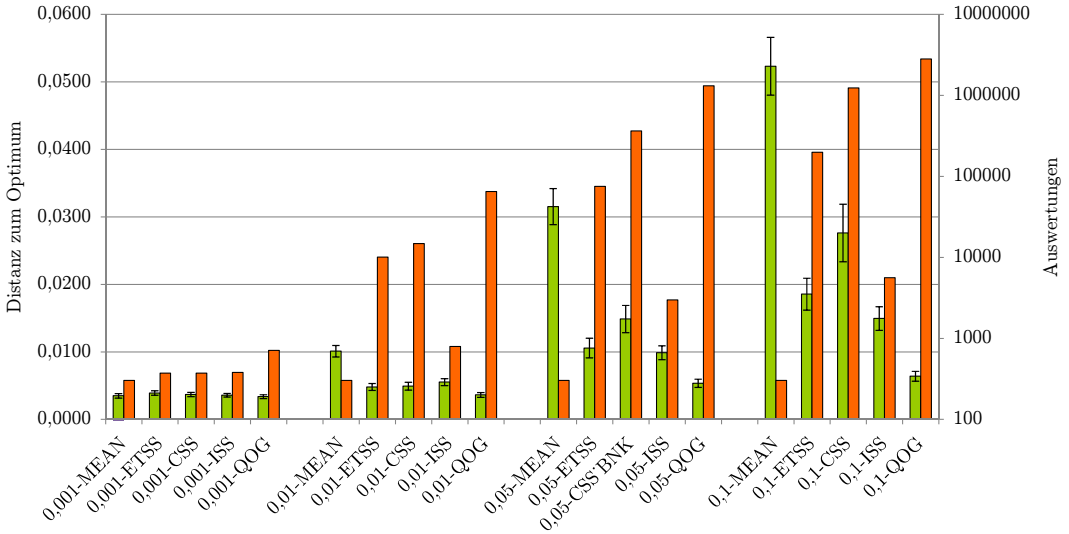



Abbildung 9.13.: Durchschnittliche Distanz zum Optimum und Anzahl Replikationen für unterschiedliche RnS-Algorithmen und Störungen am Beispiel der Ackley-Funktion und adaptivem $d^*$

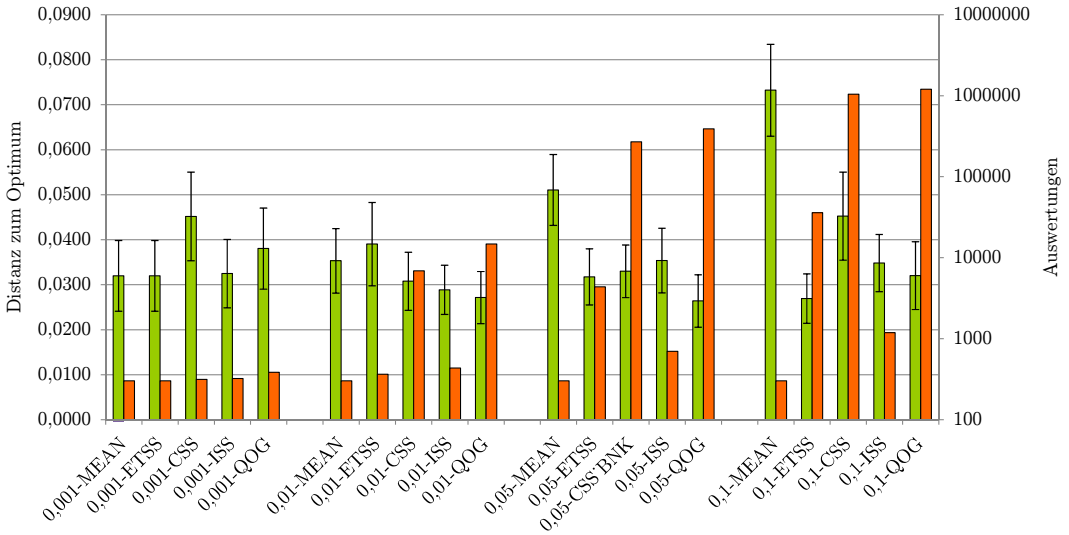



Abbildung 9.14.: Durchschnittliche Distanz zum Optimum und Anzahl Replikationen für unterschiedliche RnS-Algorithmen und Störungen am Beispiel der Schaffer-Funktion und adaptivem $d^*$

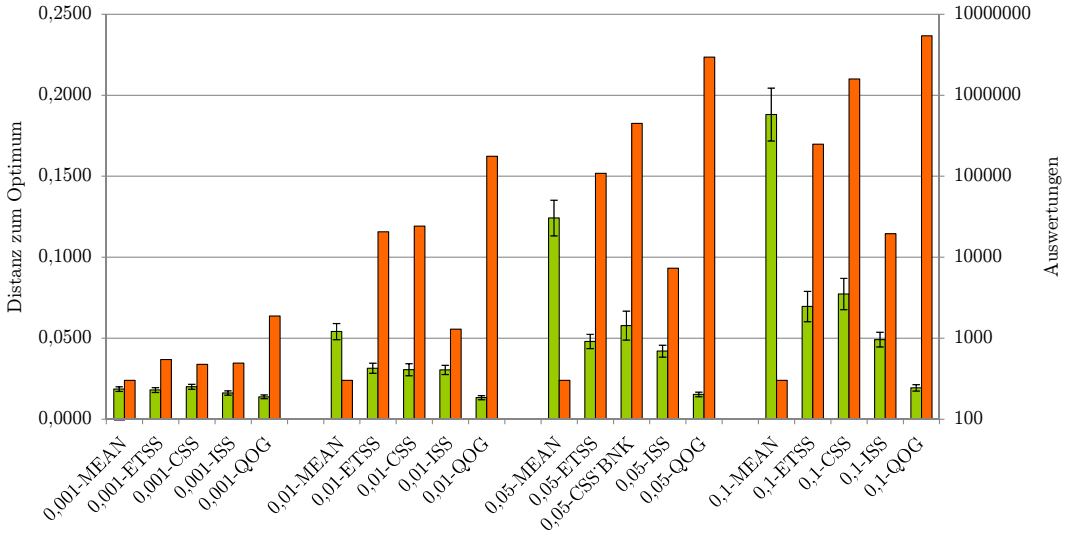



Abbildung 9.15.: Durchschnittliche Distanz zum Optimum und Anzahl Replikationen für unterschiedliche RnS-Algorithmen und Störungen am Beispiel der Sinus-Funktion mit adaptivem $d^*$ und intensiver Untersuchung

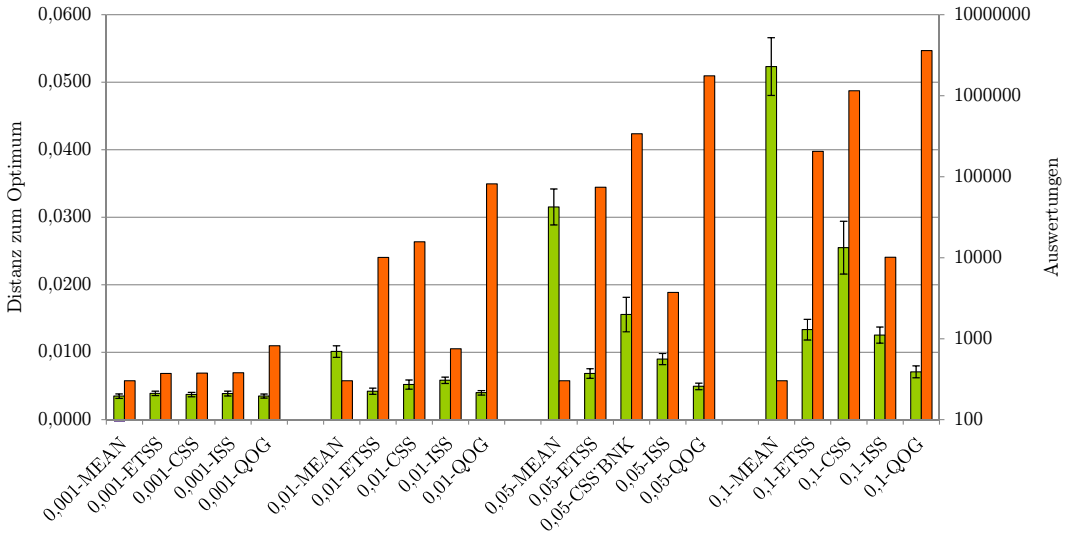



Abbildung 9.16.: Durchschnittliche Distanz zum Optimum und Anzahl Replikationen für unterschiedliche RnS-Algorithmen und Störungen am Beispiel der Ackley-Funktion mit adaptivem $d^*$ und intensiver Untersuchung

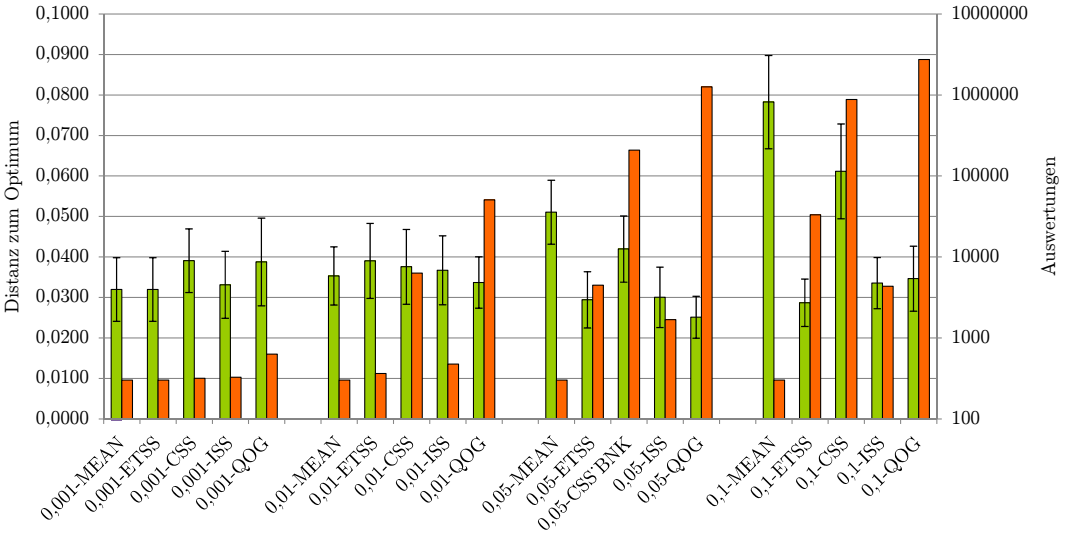



Abbildung 9.17.: Durchschnittliche Distanz zum Optimum und Anzahl Replikationen für unterschiedliche RnS-Algorithmen und Störungen am Beispiel der Schaffer-Funktion mit adaptivem $d^*$ und intensiver Untersuchung

führen dementsprechend die zu Beginn einer Optimierung eingesparten Auswertungen zu späteren Zeitpunkten durch, wenn $d^*$ entsprechend niedrige Werte erreicht hat. Durch diese spätere Auswertung wird im Allgemeinen weder eine Verbesserung noch eine Verschlechterung der Ergebnisse herbeigeführt.

## 9.4. Modifikation des Expected Improvements

Da bei normalen Kriging Metamodellen davon ausgegangen wird, dass eine deterministische Funktion zugrunde liegt, wird der erwartete Fehler $s(x)$ an bekannten Punkten als 0 angenommen. Daher bestünde hier eine direkte Ansatzmöglichkeit, diesen Schätzer anzupassen. Modifikationen dieses Schätzers in der Art, dass auch an bekannten Punkten ein Fehler vorliegen kann, führen jedoch dazu, dass z.B. das EI an bekannten Punkten größer als 0 sein kann, was jedoch in der Regel unerwünscht ist. Darüber hinaus würde dieser Ansatz zwei Arten von Fehlern vermischen: $s(x)$ trifft eine Aussage über den erwarteten Fehler, der durch die Approximation auftritt, und durch die Modifikation würde eine Unsicherheit in der Messung diesem hinzugefügt. Daher erscheint es sinnvoller, einen zweiten Schätzer $s_{K_r}$ für die Unsicherheit der Messung einzuführen, da so das KM von der Veränderung unberührt bleibt.

Der Schätzer $s_{K_r}$ berechnet sich dabei auf der Basis eines weiteren KM $K_r$. Für

jeden bekannten Punkt wird die halbe Größe des 95% Konfidenzintervalls berechnet und $K_r$ hinzugefügt. Mit dem so angepassten Metamodell $K_r$ kann der Schätzer $s_{K_r}$ als $s_{K_r}(x) = \hat{y}_{K_r}(x)$ geschrieben werden. Dieser Schätzer wird dann wie folgt in die Berechnung des EI integriert:

$$EI^*_{K_r,\alpha}(I) = s_{K_r}(x)\left(u\Phi(u) + \phi(u)\right) \tag{9.20}$$

mit

$$u = \frac{f^*_{K_r,min,\alpha} - \left(\hat{y}_{K_r}(x) - \alpha \cdot s_{K_r}(x)\right)}{s_{K_r}(x)} \tag{9.21}$$

und

$$f^*_{K_r,min,\alpha} = min_{i=1,\ldots,n}\left(f_{K_r,x_i} - \alpha \cdot s_{K_r}(x_i)\right) \tag{9.22}$$

$\alpha \in [0;1]$ ist dabei ein Steuerungsparameter, der den Einfluss von $s_{K_r}$ gewichtet. Wird $\alpha = 0$ gewählt, entspricht diese Formel dem normalen EI. Mit diesem Schätzer kann die Unsicherheit im Metamodell berücksichtigt werden, ohne die Eigenschaft $EI(x) = 0$ an bekannten Punkten $x$ zu verletzen.

Eine weitere Möglichkeit die Unsicherheit in KM innerhalb des EI zu berücksichtigen, besteht in der Erstellung eines KM der Untergrenzen der Konfidenzintervalle. Dadurch werden Bereiche stärker gewichtet, in denen Punkte liegen, deren Messwerte zu großen Konfidenzintervallen führen. Um dies zu erreichen, wird ein neues KM $KM_u$ erstellt, das alle Punkte des KM $K$ enthält. Die Funktionswerte werden jedoch so modifiziert, dass sie die Untergrenze der Konfidenzintervalle wiedergeben. Der neue Funktionswert eines Punkts $x_i$ entspricht also $y'_i = \bar{y}_i - conf(y_{i,1}, \ldots, y_{y_m})$, wobei $m$ die Anzahl der Beobachtungen am Punkt $x_i$ und $\bar{y}_i$ der Mittelwert aller Messwerte ist. Die Punkte im neuen Metamodell $KM_u$ enthalten also jeweils genau einen Messwert mit dem Wert $y'_i$. Das auf der Basis des Metamodells $KM_u$ berechnete EI wird $EI_u$ genannt. Auch hier kann ein Gewichtungsfaktor $\alpha$ genutzt werden, um den Einfluss der Größe der Konfidenzintervalle zu steuern.

Abschließend sollen diese Modifikationen mit dem normalen EI verglichen werden. Dabei werden vier Konfigurationen dem normalen EI verglichen. Neben jeweils einer Konfiguration für $EI^*$ und $EI_u$ mit $\alpha = 1$ werden zwei weitere Konfigurationen betrachtet, bei denen der Wert von $\alpha$ zu Beginn der Optimierung 1 ist und bis zum Ende auf 0 sinkt. Bei allen Experimenten wird eine normale KMO durchgeführt, die 120 Punkte dem KM hinzufügt. Die Experimente werden für jede Konfiguration 100-mal wiederholt. Da alle Verfahren die gleiche Anzahl von Auswertungen durchführen, wird für diese Experimente die durchschnittlich benötigte Zeit erfasst.

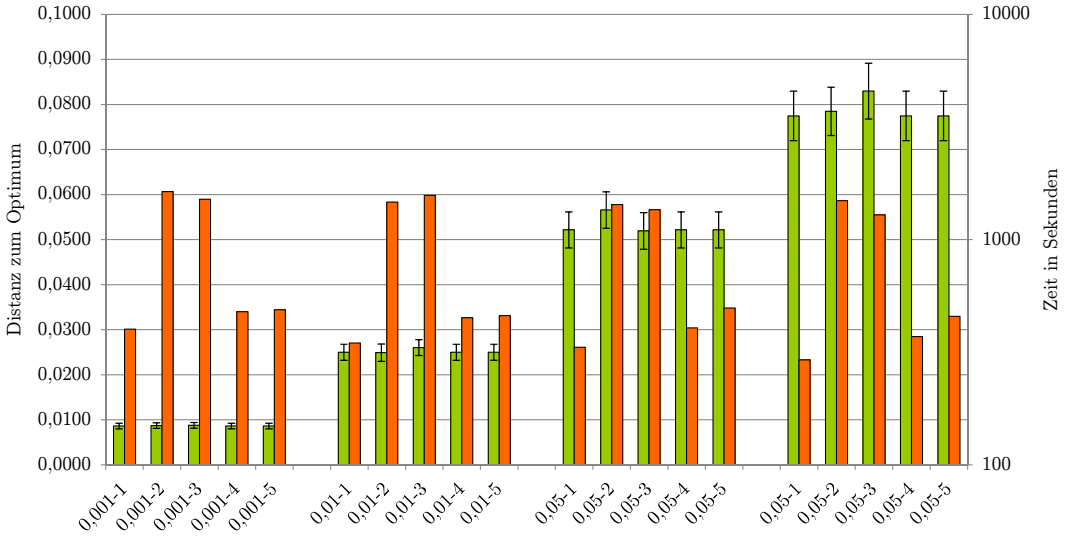



Abbildung 9.18.: Durchschnittliche Distanz zum Optimum und benötigte Zeit unterschiedliche EI am Beispiel der Sinus-Funktion

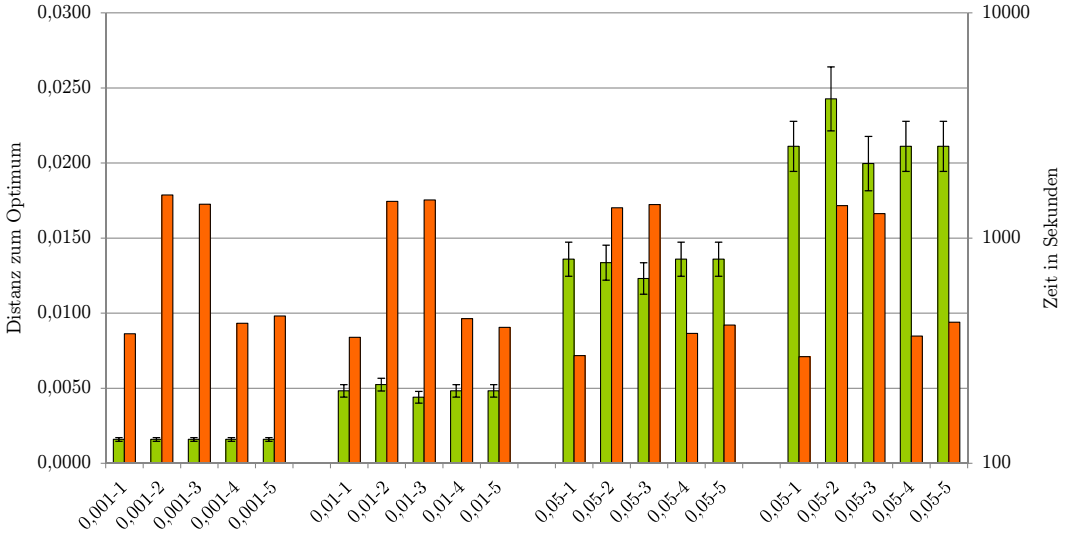



Abbildung 9.19.: Durchschnittliche Distanz zum Optimum und benötigte Zeit unterschiedliche EI am Beispiel der Ackley-Funktion

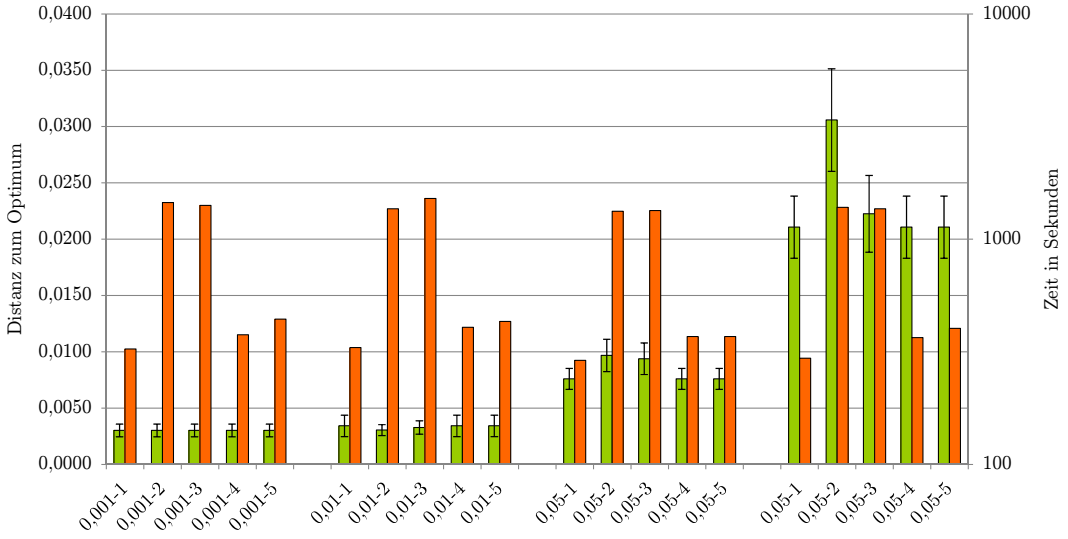



Abbildung 9.20.: Durchschnittliche Distanz zum Optimum und benötigte Zeit unterschiedliche EI am Beispiel der Schaffer-Funktion

| Nr. | EI | Dynamisch |
|---|---|---|
| 1 | EI | - |
| 2 | $EI^*$ | nein |
| 3 | $EI^*$ | ja |
| 4 | $EI_u$ | nein |
| 5 | $EI_u$ | ja |

Tabelle 9.1.: Untersuchte Konfigurationen des HKO

In den Abbildungen 9.18, 9.19 und 9.20 sind die Ergebnisse der Experimente dargestellt. Dabei steht jedes Balkenpaar für eine Konfiguration, wobei der linke hellere Balken die durchschnittliche Distanz zum Optimum mit 90% Konfidenzinterval und der rechte dunklere Balken die durchschnittlich benötigte Zeit angibt. In jeder Balkengruppe, die aus jeweils fünf Balkenpaaren besteht, wurden alle Experimente mit gleich hoher Störung durchgeführt. Von links nach rechts steigt die Störung über die Balkengruppen jedoch an. Die Reihenfolge der Konfigurationen in Tabelle 9.1 entspricht der Reihenfolge der Konfigurationen in den Balkengruppen.

Die Abbildungen 9.18, 9.19 und 9.20 zeigen deutlich, dass die beiden vorgestellten Varianten $EI^*$ und $EI_u$ in dieser Form keine Vorteile gegenüber dem normalen EI haben. Allerdings werden mit $EI_u$ in allen Fällen gleichwertige Resultate erzielt.

## 9.5. Zusammenfassung

In diesem Kapitel wurden erfolgreiche Verfahren zur dynamischen Anpassung der Anzahl der Auswertungen vorgestellt. Diese Verfahren ermöglichen, sich adaptiv der vorliegenden Situation anzupassen. Mit der Integration von RnS ist ebenfalls eine erfolgreiche Weiterentwicklung der KMO gelungen, so dass hiermit verschiedene Möglichkeiten existieren, KMO auf simulativen Modellen zu nutzen. Auch wenn die Anzahl der Auswertungen durch diese Verfahren steigt, sind diese Erweiterungen sinnvoll, da sich die Verfahren selbständig an die vorgefundene Situation anpassen und mit einer allgemeinen Konfiguration in vielen Situationen gute Ergebnisse produzieren.

Nicht erfolgreich war bisher die Entwicklung eines neuen EI, das die Störung der Messungen adäquat berücksichtigt. Die Ansätze sollten allerdings nicht verworfen werden, da diese ggf. mit anderen Verfahren, die z.B. die Unsicherheit im Metamodell reduzieren, kombiniert werden können.

# 10. Zusammenfassung und Ausblick

In dieser Arbeit wurden Suchheuristiken für ereignisdiskrete Simulationsmodelle vorgestellt. Diese Verfahren sind in vielen Bereichen einsetzbar, zu denen die Planung von Maschinen, Fabriken oder auch Logistiknetzen gehören. Die Analyse ereignisdiskreter Modelle ist ein bereits gut untersuchter, aber immer noch aktiver Forschungszweig, dessen grundlegende Verfahren in Kapitel 2 beschrieben wurden. Die in diesem Kapitel beschriebene Abstraktion der Betrachtung von Modellen mit Freiheitsgraden und einer zugeordneten Analysemethode als Funktionen ermöglicht eine verständliche Beschreibung der Methoden, mit denen eine systematische Untersuchung dieser Modelle durchgeführt werden kann. Die in Kapitel 3 eingeführten Experimentdesigns sind sowohl als eigenständige Methoden zur Betrachtung von Funktionen als auch als Teil komplexerer Verfahren einsetzbar. Dabei kann über Experimentdesigns vor allem ein Eindruck über das Verhalten der untersuchten Funktion erhalten werden, wobei dieser Eindruck abhängig vom eingesetzten Experimentdesign für einen eingeschränkten lokalen Wertebereich oder für die Funktion im gesamten Wertebereich gilt. Um aus diesem Eindruck belastbare Erkenntnisse zu erhalten, können die in Kapitel 4 vorgestellten Approximationsverfahren genutzt werden. Diese Approximationsverfahren ermöglichen - entsprechend der eingesetzten Experimentdesigns - Aussagen über das Verhalten der Funktion in einem lokalen oder den vollständigen Wertebereich, sofern die zugehörigen Annahmen erfüllt sind. Die vorgestellten Approximationsverfahren erstellen dabei Metamodelle, die als Ersatzfunktionen für meist zeitaufwändig auszuwertende Funktionen genutzt werden können. Für diese Arbeit sind die Kriging Metamodelle von besonderer Bedeutung, da sie sich durch eine hohe Flexibilität und bei deterministischen Funktionen durch eine exakte Wiedergabe bereits bekannter Punkte auszeichnen.

In Kapitel 5 wurden verschiedene bekannte Suchheuristiken vorgestellt. „Pattern Search“ und die „Response Surface Methodology“ sind lokale Verfahren, die für unimodale Funktionen entwickelt wurden. Beide Verfahren können mit ereignisdiskreten Simulationsmodellen arbeiten, wobei gegebenenfalls Erweiterungen genutzt werden müssen, die nicht Bestandteil des ursprünglichen Konzepts sind. Des weiteren wurden „Evolutionäre Algorithmen“ kurz eingeführt, da diese im Rahmen der umfangreicher betrachteten Kriging Metamodelle zur Verbesserung der Approximation eingesetzt, aber in dieser Arbeit nicht als eigenständige Verfahren untersucht werden. Basierend auf Kriging Metamodellen wurde in Kapitel 6 eine Suchheuristik für multimodale Funktionen beschrieben, die sich an dem Verfahren „Efficient

Global Optimization“ aus [JSW98] orientiert. Diese KMO genannte Suchheuristik bildet die Grundlage für die in dieser Arbeit neu eingeführten Verfahren.

In Kapitel 7 wurde eine Erweiterung der Kriging Metamodelle beschrieben, die aufgrund ihrer hierarchischen Struktur „Hierarchisches Kriging Metamodell“ genannt wird. Grundidee dieser Erweiterung ist die Gruppierung der für die Erstellung des Metamodells verwendeten Punkte nach potentiellen Extrema. Entsprechend dieser Gruppierung wird ein globales Kriging Metamodell und für jede Gruppe ein lokales Kriging Metamodell erstellt. Die Kombination der Approximationen durch die verschiedenen Metamodelle wird über Einflussfunktionen ermöglicht. Bei der Entwicklung der hierarchischen Kriging Metamodelle wurden Ansätze für die beiden gegenläufigen Ziele Laufzeitreduktion und Erhöhung der Approximationsqualität entwickelt. Die „reduzierten Kriging Metamodelle“ verringern die Laufzeit des Verfahrens, indem potentiell für die Approximation der Funktion unwichtigere Punkte nicht bei der Berechnung des Metamodells berücksichtigt werden. Bei dieser Version der hierarchischen Kriging Metamodelle werden die lokalen Kriging Metamodelle nicht berücksichtigt. Bei den restlichen hierarchischen Kriging Metamodellen ermöglichen die lokalen Metamodelle eine genauere Approximation der Funktion in den Bereichen potentieller Optima und unterstützen so die Kriging Metamodell basierte Optimierung. Durch diesen zusätzlichen Rechenaufwand erhöht sich jedoch gleichzeitig die Laufzeit des Verfahrens.

Für die Kombination der Schätzwerte von globalem und lokalen Kriging Metamodellen wurden verschiedene Funktionen untersucht, wobei sich die Methode „Point“ als besonders robust erwiesen hat. Es ist jedoch nicht auszuschließen, dass andere bisher nicht untersuchte Funktionen bessere oder zumindest für bestimmte Situationen bessere Eigenschaften mit sich bringen. Weitere Forschungsarbeit sollte daher in die Untersuchung weiterer Funktionen investiert werden. Zur Gruppierung der Punkte wird ein Standardclusterbildungsverfahren auf Basis der euklidischen Distanz eingesetzt. Kriging Metamodelle ermöglichen jedoch die Berechnung der Kriging Distanz, die den Einfluss der einzelnen Faktoren auf die Funktionswerte berücksichtigt. Eine Gruppierung entsprechend der Kriging Distanz erfordert jedoch die Erstellung eines temporären Kriging Metamodells, dass ausschließlich für die Durchführung des Clusterbildungsverfahrens genutzt wird. Eine möglicherweise erzielte Verbesserung der Qualität der Clusterbildung muss daher den zusätzlichen Rechenaufwand rechtfertigen.

Während der Optimierung basierend auf hierarchischen Kriging Metamodellen muss entschieden werden, wann eine vollständige Neuerstellung der Cluster gerechtfertigt ist. Das vorgestellte Verfahren nutzt dazu verschiedene Heuristiken, die z.B. die Anzahl der Punkte der Cluster berücksichtigen. Alternativen oder Erweiterungen dieser Heuristiken können darüber hinaus z.B. die Anzahl der noch durchzuführenden Iterationen oder die Qualität der Approximation berücksichtigen. Eine Alternative zur vollständigen Neuerstellung der Cluster ist die Verschiebung einzelner Punkte von

einem Cluster in einen anderen. Für diese Erweiterung müssten Algorithmen entwickelt werden, die die Auswahl der Punkte und Cluster durchführen. Darüber hinaus müssten Bewertungsverfahren entwickelt werden, die die Zeitpunkte einer vollständigen Neuerstellung bzw. einer Verschiebung ermitteln.

In Kapitel 8 wurde ein hybrides Verfahren basierend auf PS und KMO vorgestellt, dass die Vorteile beider Verfahren kombiniert. Während KMO in multimodalen Funktionen in kurzer Zeit Bereiche mit potentiellen Optima identifizieren kann, kann PS mit wenig Aufwand ein Optimum in unimodalen Funktionen bestimmen. Dieses Verfahren nutzt daher KMO für die globale Optimierung und PS zur Verbesserung der Ergebnisse. Es wurden verschiedene Varianten dieses Verfahrens entwickelt, die jeweils eigene Stärken und Schwächen aufweisen. Mit diesen Verfahren konnten bessere Ergebnisse als mit KMO erreicht werden, es existiert jedoch keine Variante, die in allen Fällen KMO überlegen war. Die Wahl der Variante des hybriden Verfahrens hängt daher stark von der zu untersuchenden Funktion ab. Zur Bestimmung, ob der nächste Schritt des Verfahrens global oder lokal ist, wurde eine Heuristik eingesetzt. Die Untersuchung unterschiedlicher für diesen Zweck einsetzbarer Heuristiken kann zur Entwicklung eines verbesserten hybriden Verfahrens führen, wodurch in diesem Bereich weitere Forschungen gerechtfertigt sind. Unabhängig davon kann die Kombination von anderen globalen und lokalen Suchheuristiken innerhalb dieses Verfahrens zu besseren Resultaten führen.

In Kapitel 9 wurde KMO für den Einsatz mit Simulationsmodellen angepasst. Dabei wurden verschiedene Ansätze untersucht. Der erste Ansatz verkleinert durch eine dynamische Anpassung der Replikationen für jeden Punkt die Unsicherheit in die gemessenen Werte, um auf dieser Basis sicherere Aussagen über das Verhalten der Funktion treffen zu können. Die dadurch erzielte Verbesserung führte jedoch zu einer stark angestiegenen Anzahl Replikationen, die den Einsatz für viele Simulationsmodelle aufgrund von Zeitrestriktionen verwehrt. Eine Verbesserung dieser Situation wurde durch den Einsatz von „Ranking and Selection“ Verfahren erzielt, so dass die Anzahl der Replikationen signifikant gesunken ist. Zuletzt wurden Modifikationen der Schätzer der Kriging Metamodelle vorgestellt, die es ermöglichen, die Eigenschaften von Simulationsmodellen besser zu berücksichtigen. In diesem Bereich kann eine Untersuchung bezüglich der Kombination dieser Modifikationen mit anderen bereits in der Literatur zu findenden Ansätzen zu weiteren Verbesserungen führen. Auch die Untersuchung weitere RnS-Verfahren in Kombination mit Kriging Metamodellen erscheint lohnenswert, da bei den durchgeführten Untersuchungen vor allem die neueren Verfahren zu guten Ergebnissen geführt haben.

Für eine Fortführung dieser Arbeit existieren genügend offene Fragen und Aufgabenstellungen, auch wenn bereits in dieser Arbeit deutliche Fortschritte präsentiert werden konnten. Eine direkt Fortsetzung wäre durch die Kombination der drei hier vorgestellten Bereiche gegeben. Vor allem die Kombination von hierarchischen Kriging Metamodellen mit dem hybridem Verfahren als auch die Kombination von hierarchischen Kriging Metamodellen mit den Anpassungen für Simulationsmodelle stellen

interessante und umfangreiche Forschungsbereiche dar. Darüber hinaus sind Kriging Metamodelle in den letzten Jahren von unterschiedlichen Forschungsgruppen intensiv wissenschaftlich untersucht und erweitert worden, so dass die Möglichkeit zur Kombination mit anderen Ansätzen stetig wächst.

# Anhang A.

# Berechnung von $h_q$ aus Quick Optimization through Guessing

In Abschnitt 9.3.1 wurde das Verfahren für QOG [OK07] beschrieben. Dabei blieb jedoch offen, wie die Werte für $h_q$ berechnet werden können. Diese Informationen werden im Folgenden nachgeliefert. Dabei wird die bereits in Abschnitt 9.3.1 eingeführte Ungleichung 9.16 zur einfacheren Beschreibung genutzt:

$$H(n,h,n_0) = E\left[\left(1+exp\left(\sqrt{\frac{h^2Q_1}{n_0-1}\left(1+\frac{Q_1}{Q_2}\right)}Z+\frac{h^2Q_1}{n_0-1}\right)\right)^{-1}\right] \tag{A.1}$$

Der Wert des zugrunde liegenden Integrals kann nicht direkt bestimmt werden. Daher wird zu diesem Zweck die Monte Carlo Integration genutzt (siehe [Ham60] und [PFTV92]). Die Monte Carlo Integration wird für jeden Wert mit 10000 Versuchen durchgeführt und anschließend der Mittelwert über 1000 Integrationen gebildet.

Nun kann $h_q$ für feste $n_0$, $\alpha$ und $n$ bestimmt werden. Da während der Durchführung einer Optimierung $n_0$ und $\alpha$ konstant bleiben, $n$ im Laufe der Verfahren jedoch ansteigt, ist eine Funktion $h_{n_0,\alpha}(n)$ wie folgt definiert:

$$h_{n_0,\alpha}(n) = min_{h\in\mathbb{R}}\left\{H(n,h,n_0) \leq \frac{\alpha}{n-1}\right\} \tag{A.2}$$

Die während der Tests häufig benötigten Werte für $h_{n_0,\alpha}(n)$ sind in den Tabellen A.1, A.2 und A.3 abgebildet.

Da die Berechnung von $h_{n_0,\alpha}(n)$ mittels Monte Carlo Integration entweder ungenau oder zeitaufwändig ist, werden in der vorliegenden Implementierung eine statische und eine dynamische Tabelle geführt. Die statische Tabelle enthält die am häufigsten genutzten Werte aus den Tabellen A.1, A.2 und A.3, so dass diese Werte nicht erneut berechnet werden. Die dynamische Tabelle wird als Liste geführt und enthält Werte, die mittels Monte Carlo Integration berechnet werden mussten, da sie nicht in der statischen Tabelle vorhanden sind.

| n | $h_{3;0,05}(n)$ | $f_{3;0,05}(n)$ | $(h_{3;0,05}(n) - f_{3;0,05}(n))^2$ |
|---|---|---|---|
| 3 | 7,8591 | 7,8781 | 0,0004 |
| 4 | 9,6301 | 9,6390 | 0,0001 |
| 5 | 11,1149 | 11,1154 | 0,0000 |
| 6 | 12,4208 | 12,4125 | 0,0001 |
| 7 | 13,5808 | 13,5831 | 0,0000 |
| 8 | 14,6575 | 14,6585 | 0,0000 |
| 9 | 15,6458 | 15,6588 | 0,0002 |
| 10 | 16,6191 | 16,5977 | 0,0005 |
| 11 | 17,4739 | 17,4855 | 0,0001 |
| 12 | 18,3397 | 18,3297 | 0,0001 |
| 13 | 19,1559 | 19,1361 | 0,0004 |
| 14 | 19,8958 | 19,9095 | 0,0002 |
| 15 | 20,6755 | 20,6536 | 0,0005 |
| 16 | 21,3801 | 21,3715 | 0,0001 |
| 17 | 22,1076 | 22,0658 | 0,0017 |
| 18 | 22,7424 | 22,7388 | 0,0000 |
| 19 | 23,3802 | 23,3922 | 0,0001 |
| 20 | 24,0618 | 24,0277 | 0,0012 |
| 21 | 24,6747 | 24,6468 | 0,0008 |
| 22 | 25,2078 | 25,2505 | 0,0018 |
| 23 | 25,8165 | 25,8401 | 0,0006 |
| 24 | 26,3858 | 26,4165 | 0,0009 |
| 25 | 26,9427 | 26,9805 | 0,0014 |
| 26 | 27,5207 | 27,5328 | 0,0001 |

Tabelle A.1.: $h$ aus [OK07]

Alternativ kann die Funktion $h_{n_0,\alpha}(n)$ für feste $n_0$ und $\alpha$ approximiert werden. Für $n_0$ und $\alpha$ wurde folgende Funktion mittels der Methode der kleinsten Fehlerquadrate an die bekannten Messwerte angepasst:

$$f^*_{3;0,05}(n) = a_1\sqrt{n - a_2} + b_1(n - b_2) + c_1(n - c_2)^2 + e_1 + f_1 ln(n - f_2) \qquad \text{(A.3)}$$

Das Ergebnis dieser Anpassung mit einer durchschnittlichen quadrierten Abweichung von $0,00954$ ist:

$$f_{3;0,05}(n) = 5,419824\sqrt{n - 1,095053} + 0,003809 \cdot (n - 0,625711) + 0,38864 \quad \text{(A.4)}$$

Diese Funktion wird genutzt, um für $n \leq 100$ die Werte für $h_q$ zu berechnen.

| n | $h_{3;0,05}(n)$ | $f_{3;0,05}(n)$ | $(h_{3;0,05}(n) - f_{3;0,05}(n))^2$ |
|---|---|---|---|
| 27 | 28,0736 | 28,0743 | 0,0000 |
| 28 | 28,6136 | 28,6055 | 0,0001 |
| 29 | 29,0677 | 29,1270 | 0,0035 |
| 30 | 29,5831 | 29,6393 | 0,0032 |
| 31 | 30,2068 | 30,1429 | 0,0041 |
| 32 | 30,6302 | 30,6381 | 0,0001 |
| 33 | 31,1973 | 31,1255 | 0,0052 |
| 34 | 31,6136 | 31,6054 | 0,0001 |
| 35 | 32,0832 | 32,0781 | 0,0000 |
| 36 | 32,6540 | 32,5439 | 0,0121 |
| 37 | 33,0947 | 33,0032 | 0,0084 |
| 38 | 33,5010 | 33,4561 | 0,0020 |
| 39 | 33,8021 | 33,9030 | 0,0102 |
| 40 | 34,3311 | 34,3441 | 0,0002 |
| 41 | 34,7473 | 34,7796 | 0,0010 |
| 42 | 35,2073 | 35,2098 | 0,0000 |
| 43 | 35,7042 | 35,6347 | 0,0048 |
| 44 | 36,0199 | 36,0547 | 0,0012 |
| 45 | 36,5400 | 36,4698 | 0,0049 |
| 46 | 36,9017 | 36,8803 | 0,0005 |
| 47 | 37,3265 | 37,2863 | 0,0016 |
| 48 | 37,5833 | 37,6879 | 0,0109 |
| 49 | 38,0806 | 38,0853 | 0,0000 |
| 50 | 38,4873 | 38,4787 | 0,0001 |
| 51 | 38,8936 | 38,8680 | 0,0007 |
| 52 | 39,4287 | 39,2535 | 0,0307 |
| 53 | 39,5662 | 39,6353 | 0,0048 |
| 54 | 39,9569 | 40,0135 | 0,0032 |
| 55 | 40,3326 | 40,3881 | 0,0031 |
| 56 | 40,7973 | 40,7593 | 0,0014 |
| 57 | 40,9505 | 41,1272 | 0,0312 |
| 58 | 41,4271 | 41,4918 | 0,0042 |
| 59 | 41,8020 | 41,8533 | 0,0026 |
| 60 | 42,2997 | 42,2117 | 0,0077 |
| 61 | 42,6813 | 42,5671 | 0,0130 |
| 62 | 42,7705 | 42,9196 | 0,0222 |
| 63 | 43,3715 | 43,2692 | 0,0105 |
| 64 | 43,4896 | 43,6161 | 0,0160 |
| 65 | 44,0492 | 43,9602 | 0,0079 |
| 66 | 44,0831 | 44,3017 | 0,0478 |

Tabelle A.2.: $h$ aus [OK07] (Fortsetzung)

| n | $h_{3;0,05}(n)$ | $f_{3;0,05}(n)$ | $(h_{3;0,05}(n) - f_{3;0,05}(n))^2$ |
|---|---|---|---|
| 67 | 44,7786 | 44,6406 | 0,0190 |
| 68 | 44,8919 | 44,9770 | 0,0072 |
| 69 | 45,2228 | 45,3108 | 0,0077 |
| 70 | 45,3333 | 45,6423 | 0,0955 |
| 71 | 45,9778 | 45,9714 | 0,0000 |
| 72 | 46,5251 | 46,2982 | 0,0515 |
| 73 | 46,5635 | 46,6227 | 0,0035 |
| 74 | 46,9460 | 46,9450 | 0,0000 |
| 75 | 47,4348 | 47,2651 | 0,0288 |
| 76 | 47,6765 | 47,5830 | 0,0087 |
| 77 | 47,9163 | 47,8989 | 0,0003 |
| 78 | 48,3136 | 48,2127 | 0,0102 |
| 79 | 48,4541 | 48,5246 | 0,0050 |
| 80 | 48,8106 | 48,8344 | 0,0006 |
| 81 | 49,0830 | 49,1423 | 0,0035 |
| 82 | 49,3321 | 49,4484 | 0,0135 |
| 83 | 49,6761 | 49,7525 | 0,0058 |
| 84 | 50,2248 | 50,0549 | 0,0289 |
| 85 | 50,6142 | 50,3554 | 0,0670 |
| 86 | 50,6379 | 50,6542 | 0,0003 |
| 87 | 51,0120 | 50,9512 | 0,0037 |
| 88 | 51,2511 | 51,2466 | 0,0000 |
| 89 | 51,0833 | 51,5402 | 0,2088 |
| 90 | 51,9591 | 51,8323 | 0,0161 |
| 91 | 52,3026 | 52,1227 | 0,0324 |
| 92 | 52,4569 | 52,4115 | 0,0021 |
| 93 | 52,6964 | 52,6987 | 0,0000 |
| 94 | 52,9953 | 52,9845 | 0,0001 |
| 95 | 53,2464 | 53,2687 | 0,0005 |
| 96 | 53,5539 | 53,5514 | 0,0000 |
| 97 | 53,8249 | 53,8326 | 0,0001 |
| 98 | 54,0169 | 54,1124 | 0,0091 |
| 99 | 54,4461 | 54,3908 | 0,0031 |

Tabelle A.3.: $h$ aus [OK07] (Fortsetzung)

# Anhang B.

# Informationen zu OPEDo

Am Lehrstuhl 4 der Fakultät für Informatik der Technischen Universität Dortmund wurden bereits verschiedene benutzerfreundliche Programme im Rahmen wissenschaftlicher Untersuchungen entwickelt. Zu diesen zählen z.B. das ProC/B-Toolset [BBF+02] und die APNN-Toolbox [BFKT01]. Die beiden genannten Werkzeuge sind Modellierungswerkzeuge für Prozessketten nach dem ProC/B-Paradigma bzw. Petri-Netze und stellen darüber hinaus verschiedene Analysemethoden zur Verfügung. OPEDo (**O**ptimierung and **P**erformace **E**valuation - **Do**rtmund - siehe Abbildung B.1) ist ein Werkzeug zur automatisierten Optimierung unterschiedlicher Modelle und insbesondere in der Lage, Modelle aus dem ProC/B-Toolset sowie der APNN-Toolbox zu verarbeiten. An der Implementierung von OPEDo haben bisher in alphabetischer Reihenfolge Dennis Müller, Ingo Schulz, Matthias Stöber und Dr. Axel Thümmler gearbeitet. Ideen und Hilfestellungen sowie weitere Analysatoren wurden von Prof. Dr. Peter Buchholz und Prof. Dr. Peter Kemper beigetragen.

Die ursprüngliche Implementierung umfasste nur RSM (siehe Abschnitt 5.5). Mittlerweile werden darüber hinaus die bereits beschriebenen Verfahren PS (siehe Abschnitt 5.4), EA (siehe Abschnitt 5.7) sowie KMO (siehe Abschnitt 6) mit den in den Kapiteln 7, 8 und 9 beschriebenen Ergänzungen unterstützt. Des Weiteren werden einige hier nicht weiter beschriebene Methoden unterstützt:

- Simplex-Verfahren nach Nelder und Mead (Implementiert in der GNU Scientific Library (GSL)[JG96])
- Gleichverteilte Zufallsauswahl
- Gitternetzsuche
- Randomisierte Gitternetzsuche

Während das Simplex-Verfahren nach Nelder und Mead ein weiteres „echtes“ Optimierungsverfahren darstellt, sind die letzten drei Verfahren vor allem zu Benchmarkzwecken oder die interne Verwendung - wie z.B. zur Ermittlung der Maximum Likelihood innerhalb von KMO - implementiert worden.

Das Kernprogramm nutzt Methoden der GSL [JG96] für statistische Berechnungen sowie die Generierung von Zufallszahlen. Als Zufallszahlengenerator dient Mersenne

Twister [MN98]. Neben der GSL wird als zweites Framework LAM/MPI [LSB+96] eingesetzt. Dieses Framework ermöglicht die Verteilung von Berechnungen in einem Netzwerk und wird bisher ausschließlich dazu eingesetzt, rechenintensive Auswertungen auf mehrere Computer zu verteilen. Die ersten Versionen des Kernprogramms wurden von Axel Thümmler implementiert, der die Entwicklung an Dennis Müller weitergab. Einige Anpassungen wie die Anbindung einiger externer Analysemethoden wurden von Matthias Stöber durchgeführt.

Die grafische Benutzeroberfläche von OPEDo wurde in Java implementiert. Die Entwicklung von diesem Teil des Werkzeugs wurde von Matthias Stöber begonnen und anschließend von Ingo Schulz fortgeführt. Hierbei steht neben der einfachen Benutzbarkeit auch eine weitgehende Visualisierung der Ergebnisse im Vordergrund, wobei insbesondere der Bereich empirischer Studien an Optimierungsalgorithmen thematisiert wird. Bei der Umsetzung wurden die folgenden Bibliotheken genutzt:

| Name | Informationen |
|---|---|
| FreeHep | Berechnung von Grafiken<br>http://java.freehep.org |
| l2fprod Common | Bereitstellung von GUI Elementen<br>http://l2fprod.com/common |
| JavaHelp | Erstellung und Darstellung der Hilfedatei<br>http://java.sun.com/javase/technologies/desktop/javahelp |
| JGoodies Forms | Java LayoutManager<br>http://jgoodies.com |
| JGoodies Looks | Java Look And Feel<br>http://jgoodies.com |

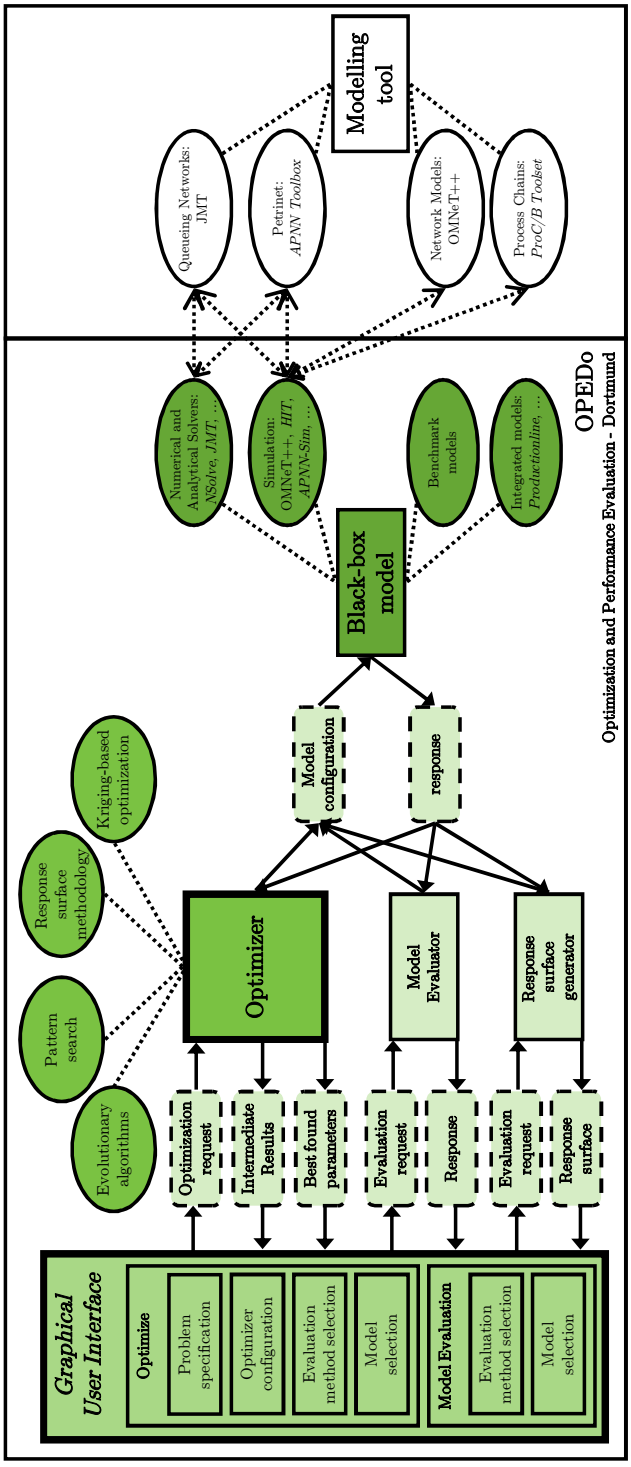



Abbildung B.1.: Übersicht von OPEDo

# Index

# Abbildungsverzeichnis

# Tabellenverzeichnis

# Pseudocodeverzeichnis

# Literaturverzeichnis


[AB02] D.V. Arnold and H.-G. Beyer. *Noisy Local Optimization with Evolution Strategies*. Kluwer Academic Publishers, Norwell, Massachusetts, USA, 2002.

[ABM09] M. Arns, P. Buchholz, and D. Müller. Optimierung ereignis-diskreter Simulationsmodelle im ProC/B-Toolset. In P. Buchholz and U. Clausen, eds., *Große Netze der Logistik*, pages 181–209. Springer, 2009.

[ANS08] B. Ankenman, B.L. Nelson, and J. Staum. Stochastic Kriging For Simulation Metamodeling. In S.J. Mason, R.R. Hill, L. Mönch, O. Rose, T. Jefferson, and J.W. Fowler, eds., *Proceedings of the 2008 Winter Simulation Conference*, pages 362–370, New York, New York, USA, 2008. Association of Computing Machinery.

[Arn06] M. Arns. *Approximative Verfahren auf erweiterten Fork/Join-Warteschlangennetzen zur Analyse von Logistiknetzen*. Dissertation, Universität Dortmund, 2006.

[Bar05] G. Bard. *Algorithms for Fast Matrix Operations*. Scholarly paper, University of Maryland, 2005.

[Bau74] H. Bauer. *Wahrscheinkeitstheorie und Grundzüge der Maßtheorie*. Walter de Gruyter & Co., 1974.

[BBF+02] F. Bause, H. Beilner, M. Fischer, P. Kemper, and M. Völker. The ProC/B Toolset for the Modelling and Analysis of Process Chains. In *Proceedings of the 12th International Conference on Computer Performance Evaluation, Modelling Techniques and Tools*, pages 51–70, London, England, 2002. Springer. LNCS 2324.

[BBK98] F. Bause, P. Buchholz, and P. Kemper. A Toolbox for Functional and Quantitative Analysis of DEDS. In R. Puigjaner, N. N. Savino, and B. Serra, eds., *TOOLS '98: 10th International Conference on Computer Performance Evaluation - Modelling Techniques and Tools*, pages 356–359, London, England, 1998. Springer.

[BBK09] F. Bause, H. Beilner, and J. Kriege. ProC/B: Eine Modellierungsumgebung zur prozessketten-orientierten Beschreibung und Analyse logistischer Netze. In P. Buchholz and U. Clausen, eds., *Große Netze der Logistik*, pages 19–57. Springer, 2009.

[BBS03] F. Bause, H. Beilner, and M. Schwenke. Semantik des ProC/B-Paradigmas. Technical Report Sonderforschungsbereich 559 „Modellierung großer Netze in der Logistik“ 03001, SFB 559, 2003.

[BD74] M.J. Box and N.R. Draper. On Minimum-Point Second-Order Designs. In *Technometrics*, volume 16, pages 613–616. American Statistical Association and American Society for Quality, 1974.

[BdCK05] B.W.M. Bettonvil, E. del Castillo, and J.P.C. Kleijnen. Statistical Testing of Optimality Conditions in Multiresponse Simulation-based Optimization. Discussion Paper 2007-45, Tilburg University, Center for Economic Research, 2005. [online; Zugriff: 13.11.2009].

[BEPW03] K. Backhaus, B. Erichson, W. Plinke, and R. Weiber. *Multivariate Analysemethoden: Eine anwendungsorientierte Einführung.* Springer, Berlin, Deutschland, 10. Auflage, 2003.

[Ber01] O. Berke. Modified Median Polish Kriging and its Application to the Wolfcamp-Aquifer Data. In *Environmetrics*, volume 12, pages 731–748. John Wiley & Sons, Ltd., 2001.

[BFKT01] P. Buchholz, M. Fischer, P. Kemper, and C. Tepper. APNN-Toolbox. http://ls4-www.cs.tu-dortmund.de/APNN-TOOLBOX, 2001. [online; Zugriff: 14.05.2009].

[BG00] H. Bandemer and A. Gebhardt. Bayesian fuzzy kriging. In *Fuzzy Sets and Systems*, volume 112, pages 405–418, Amsterdam, Niederlande, 2000. Elsevier Science B.V.

[BHH05] G.E.P. Box, J.S. Hunter, and W.G. Hunter. *Statistics for Experimenters: Design, Innovation, and Discovery.* Wiley-Interscience, 2nd edition, 2005.

[BK06] P. Buchholz and P. Kemper. Optimization of Markov Models with Evolutionary Strategies Based on Exact and Approximate Analysis Techniques. In *Proceedings 3rd International Conference on Performance Evaluation Methodology and Tools*, pages 233–242, Washington, District of Columbia, USA, 2006. IEEE Computer Society Press.

[BKvBvN07] W.E. Biles, J.P.C. Kleijnen, W.C.M. van Beers, and I. van Niewenhuyse. Kriging Metamodeling in Constrained Simulation Optimization: An Explorative Study. In S.G. Henderson, B. Biller, M.-H. Hsieh, J. Shortle, J.D. Tew, , and R.R. Barton, eds., *Proceedings of the 2007 Winter Simulation Conference*, pages 355–362. Association of Computing Machinery, 2007.

[BM09] P. Buchholz and D. Müller. Statistical Analysis and Comparison of Simulation Models of Highly Dependable Systems - An Experimental Study. In M.D. Rossetti, R.R. Hill, B. Johansson, A. Dunkin, and R.G. Ingalls, eds., *Proceedings of the 2009 Winter Simulation Conference*, volume 1, pages 516–527, Piscataway, New Jersey, USA, 2009. Institute of Electrical and Electronics Engineers. Zur Veröffentlichung akzeptiert.

[BMKT06] P. Buchholz, D. Müller, P. Kemper, and A. Thümmler. OPEDo: a tool framework for modeling and optimization of stochastic models. In *valuetools '06: Proceedings of the 1st international conference on Performance evaluation methodolgies and tools*, page 61, New York, NY, USA, 2006. Association of Computing Machinery.

[BMS05] P. Buchholz, D. Müller, and I. Schulz. OPEDo. http://ls4-www.cs.tu-dortmund.de/Opedo, 2005. [online; Zugriff: 14.05.2009].

[BMT05] P. Buchholz, D. Müller, and A. Thümmler. Optimization of Process Chain Models with Response Surface Methodology and the ProC/B Toolset. In H.-O. Günther, D. C. Mattfeld, and L. Suhl, eds., *Entscheidungsunterstützende Systeme in Supply Chain Management und Logistik*, pages 553–575. Physica-Verlag, 2005.

[BNK05] J. Boesel, B.L. Nelson, and S.H. Kim. Using Ranking and Selection to "Clean up" After Simulation Optimization. In *Operations Research*, volume 51, pages 814–825, Linthicum, Maryland, USA, 2005. Institute for Operations Research and the Management Sciences (INFORMS).

[BNN05] J. Banks, B. L. Nelson, and D. Nicol. *Discrete-Event System Simulation*. Prentice Hall International, 4th international edition, 2005.

[BSG95] R.E. Bechhofer, T.J. Santner, and D.M. Goldsman. *Design and Analysis of Experiments for Statistical Selection, Screening, and Multiple Comparisons*. John Wiley & Sons, Inc., 1995.

[BSMM97] I.N. Bronstein, K.A. Semendjajew, G. Musiol, and H. Mühlig. *Taschenbuch der Mathematik*. Verlag Harri Deutsch, Frankfurt am Main, 3. überarbeitete und erweiterte Auflage, 1997.

[BT05] P. Buchholz and A. Thümmler. Enhancing Evolutionary Algorithms with Statistical Selection Procedures for Simulation Optimization. In M.E. Kuhl, N.M. Steiger, F.B. Armstrong, and J.A. Joines, eds., *Proceedings of the 2005 Winter Simulation Conference*, pages 842–852, New York, New York, USA, 2005. Association for Computing Machinery.

[Cio02] T.M. Cioppa. *Efficient nearly orthogonal and space-filling experimental designs for high-dimensional complex models.* Dissertation, Naval Postgraduate School, Monterey, California, 2002.

[CK00] E.J. Chen and W.D. Kelton. An Enhanced Two-Stage Selection Procedure. In *Proceedings of the 2000 Winter Simulation Conference*, volume 1, pages 727–735, New York, New York, USA, 2000. Association for Computing Machinery.

[CL99] C. Cassandras and S. Lafortune. *Introduction to Discrete Event Systems.* Kluwer Academic Publishers, Dordrecht, Niederlande, 1999.

[CN80] R.D. Cook and C.J. Nachtsheim. A Comparison of Algorithms for Constructing Exact D-optimal Designs. In *Technometrics*, volume 22, pages 315–324. American Statistical Association and American Society for Quality, 1980.

[Cre93] N.A.C. Cressie. *Statistics for Spatial Data.* John Wiley & Sons, Inc., revised edition, 1993.

[dHKS06] D. den Hertog, J.P.C. Kleijnen, and A.Y.D. Siem. The Correct Kriging Variance Estimated by Bootstrapping. In *Journal of the Operational Research Society*, volume 57, pages 400–409, Basingstoke, England, 2006. Palgrave.

[DK02] D. Deflandre and J.P.C. Kleijnen. Statistical analysis of random simulations: bootstrap tutorial. Discussion Paper 2002-58, Tilburg University, Center for Economic Research, 2002. [online; Zugriff: 13.11.2009].

[DKM08] G. Dellino, J.P.C. Kleijnen, and C. Meloni. Robust Optimization in Simulation: Taguchi and Response Surface Methodology. Discussion Paper 2008-69, Tilburg University, Center for Economic Research, 2008. [online; Zugriff: 13.11.2009].

[DM05] S. D'Angelo and E.A. Minisci. Multi-Objective Evolutionary Optimization of Subsonic Airfoils by Kriging Approximation and Evolution Control. In *Proceedings of the 2005 IEEE Congress on Evolutionary Computation*, volume 2, pages 1262–1267, Edinburgh, Schotland, 2005. IEEE Service Center.

[dM06] P. di Milano. Java Modelling Tools. http://jmt.sourceforge.net/, 2006. [online; Zugriff: 14.05.2009].

[Dyk71] O. Dykstra. The Augmentation of Experimental Data to Maximize $\left|(X'X)^{-1}\right|$. In *Technometrics*, volume 13, pages 682–688, Minneapolis, Minnesota, USA, August 1971. American Statistical Association and American Society for Quality.

[Fed72] V. V. Federov. *Theory of Optimal Experiments (Probability and mathematical statistics).* Academic Press Inc, 1972.

[FK09] A.I.J. Forrester and A.J. Keane. Recent advances in surrogate-based optimization. *Progress in Aerospace Sciences*, 45:50–79, January 2009.

[FM68] A.V. Fiacco and G.P. McCormick. *Nonlinear programming: sequential unconstrained minimization techniques.* Wiley New York, 1968.

[GCR08] H. Gunes, H.E. Cekli, and U. Rist. Data Enhancement, Smoothing, Reconstruction And Optimization By Kriging Interpolation. In S. J. Mason, R.R. Hill, L. Mönch, O. Rose, T. Jefferson, and J.W. Fowler, eds., *Proceedings of the 2008 Winter Simulation Conference*, pages 379–386, New York, New York, USA, 2008. Association of Computing Machinery.

[GKBM96] M.A. Gallagher, Jr. K.W. Bauer, and P.S. Maybeck. Initial data truncation for univariate output of discrete-event simulations using the Kalman filter. *Management Science*, 42:559–575, 1996.

[God07] M. Godzierz. *Globale Optimierung extrem aufwendiger Funktionen mit hochparallelen und sequentiellen Methoden.* Dissertation, Technische Universität Darmstadt, 2007.

[HA05] D. Huang and T. Allen. Design and analysis of variable fidelity experimentation applied to engine valve heat treatment process design. In *The Journal of the Royal Statistical Society: Series C (Applied Statistics)*, volume 54, pages 443–463, Columbus, USA, 2005. Royal Statistical Society.

[Ham60] J.M. Hammersley. Monte Carlo Methods for Solving Multivariable Problems. In *Annals of the New York Academy of Sciences*, volume 86, pages 844–874. Oxford University, Oxford, England, 1960.

[HANZ05] D. Huang, T.T. Allen, W.I. Notz, and N. Zheng. Efficient optimization design method using kriging model. In *Journal of aircraft*, volume 42, pages 413–420. American Institute of Aeronautics and Astronautics, 2005.

[HANZ06] D. Huang, T.T. Allen, W.I. Notz, and N. Zheng. Global Optimization of Stochastic Black-Box Systems via Sequential Kriging Meta-Models. In *Journal of Global Optimization*, volume 34, pages 441–446, Hingham, MA, USA, 2006. Kluwer Academic Publishers.

[Hav95] B.R. Haverkort. Approximate Analysis of Networks of PH|PH|1|K Queues: Theory & Tool Support. In *MMB '95: Proceedings of the 8th International Conference on Modelling Techniques and Tools for Computer Performance Evaluation*, pages 239–253, London, England, 1995. Springer.

[HS93] R.H. Hardin and N.J.A. Sloane. A New Approach to the Construction of Optimal Designs. In *Journal of Statistical Planning and Inference*, volume 37, pages 339–369, Amsterdam, Niederlande, December 1993. Elsevier Science B.V.

[HS07] G. Hawe and J. Sykulski. Considerations of accuracy and uncertainty with kriging surrogate models in single-objective electromagnetic design optimization. In *IET Science, Measurement & Technology*, volume 1, pages 37–47. IEEE Computer Society Press, 2007.

[HS08] G.I. Hawe and J.K. Sykulski. Probability of Improvement Methods for Constrained Multi-Objective Optimization. In *The IET 7th International Conference on Computation in Electromagnetics*, pages 50–51. IEEE Computer Society Press, April 2008.

[Hua05] D. Huang. *Experimental planning and sequential Kriging optimization using multiple fidelity evaluations*. Dissertation, Ohio State University, 2005.

[JG96] G. Jungman and Dr. B. Gough. GNU Scientific Library Homepage. http://www.gnu.org/software/gsl, 1996. [online; Zugriff: 14.05.2009].

[JHS08] V.R. Joseph, Y. Hung, and A. Sudjianto. Blind kriging: A new method for developing metamodels. In *Journal of Mechanical Design*, volume 130. American Society of Mechanical Engineers, 2008.

[Jon01] D.R. Jones. A taxonomy of global optimization methods based on response surfaces. *Journal of Global Optimization*, 21(4):345–383, December 2001.

[Jos05] V.R. Joseph. *Limit Kriging*. Dissertation, School of Industrial and Systems Engineering, 2005.

[JSW98] D.R. Jones, M. Schonlau, and W.J. Welch. Efficient Global Optimization of Expensive Black-Box Functions. In *Journal of Global Optimization*, volume 13, pages 455–492, Hingham, MA, USA, 1998. Kluwer Academic Publishers.

[KBP06] J.P.C. Kleijnen, B.W.M. Bettonvil, and F. Persson. Finding the Important Factors in Large Discrete-Event Simulation. In A.M. Dean and S.M. Lewis, eds., *Screening: Methods for Experimentation in Industry, Drug Discovery, and Genetics*, pages 287–301, New York, New York, USA, 2006. Springer.

[KD06] J.P.C. Kleijnen and D. Deflandre. Validation of regression metamodels in simulation: Bootstrap approach. In *European Journal of Operational Research*, volume 170, pages 120–131, Amsterdam, Niederlande, 2006. Elsevier Science B.V.

[KdHA04] J.P.C. Kleijnen, D. den Hertog, and E. Angün. Response surface methodology's steepest ascent and step size revisited. In *European Journal of Operational Research*, volume 159, pages 121–131, Amsterdam, Niederlande, 2004. Elsevier Science B.V.

[KIL06] H.-K. Kim, C.-H. Im, and D.A. Lowther. An Optimization Framework Using Sequential Approximation Model and Multimodal Evolution Strategy. In *12th Biennial IEEE Conference on Electromagnetic Field Computation*, pages 127–127. IEEE Computer Society Press, 2006.

[Kit86] P.K. Kitanidis. Parameter uncertainty in estimation of spatial functions: Bayesian analysis. In *Water Resources Research*, volume 22, pages 499–507. American Geophysical Union, 1986.

[Kle06] J.P.C. Kleijnen. Generalized response surface methodology : a new metaheuristic. Discussion Paper 77, Tilburg University, Center for Economic Research, 2006. [online; Zugriff: 13.11.2009].

[Kle07a] J.P.C. Kleijnen. Kriging Metamodeling in Simulation: A Review. Discussion Paper 2007-13, Tilburg University, Center for Economic Research, 2007. [online; Zugriff: 13.11.2009].

[Kle07b] J.P.C. Kleijnen. Regression Models and Experimental Designs: A Tutorial for Simulation Analysts. In S.G. Henderson, B. Biller, M.-H. Hsieh, J. Shortle, J.D. Tew, , and R.R. Barton, eds., *Proceedings of the 2007 Winter Simulation Conference*, pages 183–194. Association of Computing Machinery, 2007.

[Kle07c] J.P.C. Kleijnen. Simulation Experiments in Practice: Statistical Design and Regression Analysis. In *Journal of Simulation*, volume 2, pages 19–27, Basingstoke, England, March 2007. Palgrave Macmillan.

[KMT06] P. Kemper, D. Müller, and A. Thümmler. Combining response surface methodology with numerical methods for optimization of markovian models. *IEEE Trans. Dependable Sec. Comput.*, 3(3):259–269, 2006.

[KP99] W.S. Kerwin and J.L. Prince. The Kriging Update Model and Recursive Space-Time Function Estimation. In *Transactions on Signal Processing*, volume 47, pages 2942–2952. IEEE Computer Society Press, November 1999.

[Kri51] D.G. Krige. A statistical approach to some mine valuations and allied problems at the Witwatersrand. Master's thesis, University of Witwatersrand, 1951.

[KS00] J.P.C. Kleijnen and R.G. Sargent. A methodology for fitting and validating metamodels in simulation. In *European Journal of Operational Research*, volume 120, pages 14–29, Amsterdam, Niederlande, January 2000. Elsevier Science B.V.

[Kuh95] A. Kuhn. *Prozessketten in der Logistik - Entwicklungstrends und Umsetzungsstrategien.* Verlag Praxiswissen, Dortmund, Deutschland, 1995.

[Kuh98] A. Kuhn. Sonderforschungsbereich 559 - Modellierung großer Netze in der Logistik - Universität Dortmund. http://www.sfb559.tu-dortmund.de, 1998. [online; Zugriff: 18.08.2009].

[KV08] J. Kriege and S. Vastag. ProC/B goes OMNeT++: Efficient Simulation of Process Chains. In F. Bause and P. Buchholz, eds., *14th GI/ITG Conference: Measurement, Modelling and Evaluation of Computer and Communication Systems*, pages 303–305. VDE Verlag GmbH, 2008.

[KvBvN08] J.P.C. Kleijnen, W.C.M. van Beers, and I. van Nieuwenhuyse. Constrained Optimization in Simulation: A Novel Approach. Discussion Paper 2008-95, Tilburg University, Center for Economic Research, 2008. [online; Zugriff: 13.11.2009].

[LK99] A.M. Law and W.D. Kelton. *Simulation modeling and analysis.* McGraw-Hill Higher Education, 3rd edition, 1999.

[LS05] R. Li and A. Sudjianto. Analysis of Computer Experiments Using Penalized Likelihood in Gaussian Kriging Models. In *Technometrics*, volume 47, pages 111–121, Minneapolis, Minnesota, May 2005. American Society for Quality.

[LSB+96] Dr. A. Lumsdaine, Dr. J. Squyres, B. Barrett, P. Kambadur, and J. Hursey. LAM/MPI Parallel Computing. http://www.lam-mpi.org, 1996. [online; Zugriff: 16.07.2009].

[LTT00] Robert Michael Lewis, Virginia Torczon, and Michael W. Trosset. Direct search methods: then and now. In *Journal of Computational and Applied Mathematics*, volume 124, pages 191–207. Springer, 2000.

[LZGS84] E.D. Lazowska, J. Zahorjan, G.S. Graham, and K.C. Sevcik. *Quantitative System Performance, Computer System Analysis Using Queuing Network Models: Computer Analysis Using Queuing Network Models.* Prentice Hall, Englewood Cliffs, New Jersey, USA, 1984.

[Mar06] A. Markov. Rasprostranenie zakona bol'shih chisel na velichiny, zavisyaschie drug ot druga. In *Izvestiya Fiziko-matematicheskogo obschestva pri Kazanskom universitete*, volume 2, pages 135–156, 1906.

[Mar71] A. Markov. Extension of the limit theorems of probability theory to a sum of variables connected in a chain. In R. Howard, ed., *Dynamic Probabilistic Systems (Volume I: Markov Models)*, chapter Appendix B, pages 552–577. John Wiley & Sons, Inc., New York City, 1971.

[Mat71] G. Matheron. *The Theory of Regionalized Variables and its Applications*, volume 5. Centre for Mathematical Morphology, Paris School of Mines, Fontainbleau, Frankreich, 1971.

[MBC79] M.D. McKay, R.J. Beckman, and W.J. Conover. A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code. In *Technometrics*, volume 21, pages 239–246. American Statistical Association and American Society for Quality, 1979.

[MM70] T.J. Mitchell and F.L. Miller. Use of Design Repair to Construct Designs for Special Linear Models. In *Mathematical Division Annual Progress Report*, volume 4661, Oak Ridge, Tennessee, USA, 1970. Oak Ridge National Laboratory.

[MM02] R.H. Myers and D.C. Montgomery. *Response Surface Methodology: Process and Product Optimization Using Designed Experiments*. John Wiley & Sons, Inc., 2002.

[MN98] M. Matsumoto and T. Nishimura. Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. In *Transactions on Modeling and Computer Simulation*, volume 8, pages 3–30, New York, New York, USA, 1998. Association of Computing Machinery.

[MT02] C. Mazzetti and E. Todini. Development and application of the block kriging technique to rain-gauge data. Technical report, University of Bologna, Department of Earth and Geo-Environmental Sciences, Bologna, Italy, 2002. [online; Zugriff: 13.11.2009].

[NP98] V. Nissen and J. Propach. On the Robustness of Population-Based Versus Point-Based Optimization in the Presence of Noise. In *IEEE Transactions on Evolutionary Computation*, volume 2, pages 107–119. IEEE Computer Society Press, 1998.

[NvOPD00] H.G. Neddermeijer, G.J. van Oortmarssen, N. Piersma, and R. Dekker. A Framework for Response Surface Methodology for Simulation Optimization. In J.A. Joines, R.R. Barton, K. Kang, and P.A. Fishwick, eds., *Proceedings of the 2000 Winter Simulation Conference*, pages 129–136, New York, New York, USA, 2000. Association for Computing Machinery.

[OK07] T. Osogami and S. Kato. Optimizing system configurations quickly by guessing at the performance. In *Proceedings of the 2007 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, volume 35, pages 145–156, New York, New York, USA, 2007. Association for Computing Machinery.

[PFTV92] W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling. *Numerical Recipes in FORTRAN 77: The Art of Scientific Computing.* Cambridge University Press, Cambridge, England, 2nd edition, 1992.

[Poh94] H. Pohlheim. The Genetic and Evolutionary Algorithm Toolbox. http://www.geatbx.com, 1994. [online; Zugriff: 20.07.2009].

[PT71] H.D. Patterson and R. Thompson. Recovery of inter-block information when block sizes are unequal. In *Biometrika*, volume 58, pages 545–554, 1971.

[Rat99] A. Ratle. Optimal sampling strategies for learning a fitness model. In P.J. Angeline, Z. Michalewicz, M. Schoenauer, X. Yao, and Z. Ali, eds., *Proceedings of the 1999 Congress on Evolutionary Computation*, volume 3, pages 2078–2085. IEEE Computer Society Press, 1999.

[RBL93] D. Ramachandran, E.J. Berbari, and P. Lander. Comparison of Interpolation Methods Used in Epicardial Activation Map Formation. In *Proceedings of Computers in Cardiology*, pages 129–132. IEEE Computer Society Press, September 1993.

[Rin78] Y. Rinott. On Two-Stage Selection Procedures and Related Probability-Inequalities. In *Communications in Statistics Theory and Methods*, volume 7, pages 799–811. Taylor & Francis, 1978.

[Roc] Rockwell Software. Arena. http://www.arenasimulation.com. [online; Zugriff: 14.05.2009].

[Röt97] A. Rötting. *Gradientenkriging - eine integrierende geostatische Methode zur einheitlichen Auswertung von absoluten und relativen Messdaten.* Dissertation, Technische Universität Bergakademie Freiberg, 1997.

[Sal98] R. Salomon. Evolutionary Algorithms and Gradient Search: Similarities and Differences. In *IEEE Transactions on Evolutionary Computation*, volume 2, pages 45–55. IEEE Computer Society Press, 1998.

[Sas02] M.J. Sasena. *Flexibility and Efficiency Enhancements for Constrained Global Design Optimization with Kriging Approximations.* Dissertation, University of Michigan, 2002.

[SBB06] L.M. Sanchez-Brea and E. Berabeu. Uncertainty Estimation by Convolution Using Spatial Statistics. In *Transactions On Image Processing*, volume 15, pages 3131–3137. IEEE Computer Society Press, October 2006.

[Sch95] H.-P. Schwefel. *Evolution and Optimum Seeking.* John Wiley & Sons, Inc., New York, New York, USA, 1995.

[Sri04] T.A. Sriver. *Pattern Search Ranking and Selection Algorithms for Mixed-Variable Optimization of Stochastic Systems.* Doctoral thesis, Air Force Institute Of Technology, Wright Patterson AFB, Ohio, USA, 2004.

[SSV+04] E.S. Siah, M. Sasena, J.L. Volakis, P.Y. Papalambros, and R.W. Wiese. Fast Parameter Optimization of Large-Scale Electromagnetic Objects Using DIRECT with Kriging Metamodeling. In *Transactions on Microwaves Theory and Techniques*, volume 52, pages 276–285. IEEE Computer Society Press, January 2004.

[Ste94] W.J. Stewart. *Introduction to the numerical solution of Markov chains.* Princeton Univ. Press, Princeton, New Jersey, USA, 1994.

[SWMW89] J. Sacks, W. Welch, T.J. Mitchell, and H.P. Wynn. Design and Analysis of Computer Experiments. In *Statistical Science*, volume 4, pages 409–435, 1989.

[SWN03] T.J. Santner, B.J. Williams, and W.I. Notz. *The Design and Analysis of Computer Experiments.* Springer, 2nd revised edition, 2003.

[Toc93] F. Tochu. A Contouring Program Based on Dual Kriging Interpolation. In *Engineering with Computers*, volume 9, pages 160–177, London, England, September 1993. Springer.

[Tod01] E. Todini. Influence of parameter estimation uncertainty in Kriging: Part 1 - The Theoretical Development. In *Hydrology and Earth System Sciences*, volume 5, pages 215–223, 2001. http://www.hydrol-earth-syst-sci.net/5/215/2001, [online; Zugriff: 13.11.2009].

[Tor97] V. Torczon. On the Convergence of Pattern Search Algorithms. In *Journal on Optimization*, volume 7, pages 1–25, Philadelphia, Pennsylvania, USA, 1997. Society for Industrial and Applied Mathematics.

[uYD93] B. Baynat und Y. Dallery. A unified view of product-form approximation techniques for general closed queueing networks. In *Performance Evaluation*, volume 18, pages 205–224, Amsterdam, Niederlande, 1993. Elsevier Science Publishers B. V.

[Var] A. Varga. OMNeT++. http://www.omnetpp.org. [online; Zugriff: 14.05.2009].

[vB05a] W.C.M. van Beers. *Kriging Metamodeling for Simulation*. Proefschrift, Universiteit van Tilburg, 2005.

[vB05b] W.C.M. van Beers. Kriging Metamodeling in Discrete-Event Simulation: An Overview. In M.E. Kuhl, N.M. Steiger, F.B. Armstrong, and J.A. Joines, eds., *Proceedings of the 2005 Winter Simulation Conference*, pages 202–208, New York, New York, USA, December 2005. Association of Computing Machinery.

[vBK03] W.C.M. van Beers and J.P.C. Kleijnen. Kriging for interpolation in random simulation. In *Journal of the Operational Research Society*, volume 54, pages 255–262, Basingstoke, England, 2003. Palgrave Macmillan.

[vBK08] W.C.M. van Beers and J.P.C. Kleijnen. Customized sequential designs for random simulation experiments: Kriging metamodeling and bootstrapping. In *European Journal of Operational Research*, volume 186, pages 1099–1113, Amsterdam, Niederlande, 2008. Elsevier Science B.V.

[vdKS06] J. van de Kassteele and A. Stein. A model for external drift kriging with uncertain covariates applied to air quality measurements and dispersion model output. In *Environmetrics*, volume 17, pages 309–322. John Wiley & Sons, Ltd., 2006.

[VVW07] J. Vellemonteix, E. Vazquez, and E. Walter. Identification of expensive-to-simulate parametric models using Kriging and Stepwise Uncertainty Reduction. In *Proceedings of the 46th IEEE Conference on Decision and Control*, pages 5505–5510, Washington, District of Columbia, USA, December 2007. IEEE Computer Society Press.

[VW05] E. Vazquez and E. Walter. Estimating derivatives and integrals with kriging. In *Proceedings of the 44th IEEE Conference on Decision and Control and the European Control Conference*, pages 8156–8161. IEEE Computer Society Press, December 2005.

[WBJS03] L. Willmes, T. Bäck, Y. Jin, and B. Sendhoff. Comparing Neural Networks and Kriging for Fitness Approximation in Evolutionary Optimization. In *Proceedings of the 2003 IEEE Congress on Evolutionary Computation*, volume 1, pages 663–670. IEEE Computer Society Press, December 2003.

[Wei07] K. Weicker. *Evolutionäre Algorithmen*. Karsten Teubner B.G. GmbH, 2. Ausgabe, September 2007.

[Wel81] P.D. Welch. On the Problem of the Initial Transient in Steady-State Simulation. Technical report, IBM Watson Research Center, Yorktown Heights, New York, USA, 1981.

[WM97] D.H. Wolpert and W.G. Macready. No Free Lunch Theorems for Optimization. In *IEEE Transactions on Evolutionary Computation*, volume 1, pages 67–82. IEEE Computational Intelligence Society, April 1997.

[WS07] H. Wu and F. Sun. Adaptive Kriging control of discrete-time nonlinear systems. In *Control Theory & Applications*, volume 1, pages 646–656. Institution of Engineering and Technology, May 2007.

[Wu86] Z. Wu. *Die Kriging-Methode zur Lösung mehrdimensionaler Interpolationsprobleme*. Dissertation, Georg-August-Universität zu Göttingen, 1986.

[Wyn72] H.P. Wynn. Results in the theory and construction of D-optimum experimental designs. In *Journal of the Royal Statistical Society: Series B*, volume 34, pages 133–147, Columbus, USA, 1972. Royal Statistical Society.

[Ye98a] K.Q. Ye. Column orthogonal latin hypercubes and their application in computer experiments. In *Journal of the Americal Statistical Association*, volume 93, pages 1430–1439. American Statistical Association and American Society for Quality, 1998.

[Ye98b] K.Q. Ye. On the structure of orthogonal latin hypercubes. Manuskript, Department of Statistics, University of Michigan, 1998. [online; Zugriff: 13.11.2009].

[YLS00] K.Q. Ye, W. Li, and A. Sudjianto. Algorithmic Construction of Optimal Symmetric Latin Hypercube Designs. In *Journal of Statistical Planning and Inference*, volume 90, pages 145–159, Amsterdam, Niederlande, 2000. Elsevier Science B.V.

[YNN08] J. Yin, S.H. Ng, and K.M. Ng. Kriging Model with Modified Nugget Effect for Random Simulation with Heterogeneous Variances. In M. Xie and R. Jiao, eds., *International Conference on Industrial Engineering and Engineering Management*, pages 1714–1718. IEEE Engineering Management Society, December 2008.