

**Diplomarbeit**

**Auswahl und Anpassung von  
Optimierungsmethoden zur  
simulationsbasierten Analyse  
der Übertragungssicherheit  
von Kfz-Bussystemen**

**Manuel Beer**

**8. Oktober 2009**

interne berichte  
internal reports

Betreuer:

Prof. Dr.-Ing. Stephan Frei

Prof. Dr.-Ing. Olaf Spinczyk

Arbeitsgebiet Bordsysteme

Fakultät für Elektrotechnik und

Informationstechnik

Technische Universität Dortmund

<http://www.bordsysteme.tu-dortmund.de>





# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Motivation und Hintergrund . . . . .	1
1.2	Aufbau der Arbeit . . . . .	3
<b>2</b>	<b>Analyse von Kfz-Bussystemen</b>	<b>5</b>
2.1	Grundlagen Kfz-Bussysteme . . . . .	5
2.1.1	Das FlexRay-Bussystem . . . . .	8
2.2	Analyseumgebung . . . . .	12
2.2.1	Bildung des Simulationsmodells . . . . .	13
2.2.2	Zuweisen der Parameterwerte . . . . .	13
2.2.3	Simulation des Modells . . . . .	14
2.2.4	Analyse der Signale . . . . .	14
2.2.5	Parametervariation . . . . .	15
<b>3</b>	<b>Optimierungsalgorithmen</b>	<b>17</b>
3.1	Grundlagen . . . . .	19
3.1.1	Optimierungsprobleme . . . . .	19
3.1.2	Suchraum . . . . .	19
3.1.3	Zielfunktion . . . . .	19
3.2	Hill-Climbing . . . . .	20
3.3	Tabu-Suche . . . . .	21
3.4	Simulated Annealing . . . . .	23
3.4.1	Der Algorithmus . . . . .	24
3.4.2	Anpassungen und Erweiterungen . . . . .	25
3.5	Ameisenkolonieoptimierung . . . . .	26
3.5.1	Formale Anforderungen an die Problemstruktur . . . . .	27
3.5.2	Schematischer Ablauf einer Optimierung . . . . .	27
3.6	Genetische Algorithmen . . . . .	29
3.6.1	Grundlagen . . . . .	29
3.6.2	Terminologie . . . . .	30

3.6.3	Der Algorithmus . . . . .	31
3.6.4	Operatoren . . . . .	34
3.6.5	Weitere Anpassungsmöglichkeiten . . . . .	40
3.7	Partikel Schwarm Optimierung . . . . .	42
3.7.1	Grundlagen . . . . .	42
3.7.2	Der Algorithmus . . . . .	43
3.7.3	Berechnung der Partikelbewegungen . . . . .	44
3.7.4	Trägheitsgewicht . . . . .	45
3.7.5	Anpassungen . . . . .	46
3.8	Bewertung und Auswahl der Verfahren . . . . .	47
<b>4</b>	<b>Anpassen der Algorithmen</b>	<b>49</b>
4.1	Testmodelle . . . . .	49
4.1.1	Modell 1 . . . . .	49
4.1.2	Modell 2 . . . . .	51
4.2	Anpassen des Simulated Annealing Algorithmus . . . . .	53
4.2.1	Parameterauswahl und Variation . . . . .	53
4.2.2	Cooling-Schedule . . . . .	53
4.2.3	Wahl der Abbruchparameter . . . . .	54
4.2.4	Initiale Analyse des Simulated Annealing . . . . .	54
4.2.5	Variation Cooling-Schedule . . . . .	55
4.2.6	Lokalisierungsschritte . . . . .	59
4.3	Anpassen des genetischen Algorithmus . . . . .	64
4.3.1	Abbruchkriterien . . . . .	65
4.3.2	Fitness-Skalierung . . . . .	65
4.3.3	Elite-Übernahme . . . . .	65
4.3.4	Populationsgröße . . . . .	65
4.3.5	Wahl des Selektions- und Crossover-Operators . . . . .	69
4.3.6	Verhältnis zwischen Crossover- und Mutationsrate . . . . .	75
4.4	Anpassen des PSO-Algorithmus . . . . .	78
4.4.1	Abbruchkriterien . . . . .	79
4.4.2	Initialisierung der Partikel . . . . .	79
4.4.3	Populationsgröße . . . . .	79
4.4.4	Trägheitsgewichtung . . . . .	83
4.4.5	Beschleunigungskoeffizienten . . . . .	87
4.4.6	Umgang mit Wertebereichsgrenzen . . . . .	88
4.5	Vergleich der Ergebnisse . . . . .	91
4.5.1	Modell 1 . . . . .	91
4.5.2	Modell 2 . . . . .	92

<i>INHALTSVERZEICHNIS</i>	iii
4.5.3 Auswertung . . . . .	93
<b>5 Anwendung des Analyseverfahrens</b>	<b>97</b>
5.1 Bewertung mittels Augendiagramm . . . . .	98
5.2 Signalintegritätsanalyse . . . . .	100
<b>6 Zusammenfassung und Ausblick</b>	<b>105</b>
<b>A Weitere Ergebnisse der Anpassung des SA</b>	<b>107</b>
<b>B Weitere Ergebnisse der Anpassung des GA</b>	<b>117</b>
<b>C Weitere Ergebnisse der Anpassung der PSO</b>	<b>125</b>
<b>D Weitere Ergebnisse der Signalintegritätsanalyse</b>	<b>133</b>
<b>Abbildungsverzeichnis</b>	<b>138</b>
<b>Tabellenverzeichnis</b>	<b>140</b>
<b>Literaturverzeichnis</b>	<b>144</b>
<b>Erklärung</b>	<b>144</b>



# 1 Einleitung

## 1.1 Motivation und Hintergrund

Die Ansprüche, die an neue Fahrzeuge gestellt werden, wachsen und verändern sich stetig. Fahrzeuge sollen heutzutage immer sicherer, komfortabler, umweltverträglicher und zuverlässiger sein. Um die vielfältigen und komplexen Anforderungen erfüllen zu können, wird in modernen Fahrzeugen eine stetig wachsende Anzahl an unterschiedlichen Steuergeräten (engl.: electronic control unit, ECU) verbaut, die die Steuerung der vielen notwendigen Systeme übernehmen. Bereits 2002 wurden etwa 70 solcher Steuergeräte in einem Oberklassefahrzeug verbaut [GBF<sup>+</sup>02]. Inzwischen dürfte dieser Wert angesichts zahlreicher Neuerungen, vor allem im Bereich Fahrerassistenzsysteme (autom. Abstandsregelung, Spurhalte- und Spurwechselassistenten), Komfort und Infotainment noch einmal deutlich angewachsen sein.

Vielfach agieren die einzelnen Systeme jedoch nicht komplett autark, sondern sind oftmals auf Informationen aus anderen Systemen angewiesen, um ihre Aufgaben erfüllen zu können. So ist beispielsweise das Zusammenspiel elektronischer Stabilitätssysteme (ESP), Bremssysteme (ABS, BAS) und Fahrwerkssteuerungen ebenso stark interagierend ausgelegt, wie bei Navigations-, Kommunikations- und Entertainmentsystemen.

Aus dieser Tatsache resultiert zum einen ein extrem hoher Verkabelungsaufwand um die Kommunikation der vielen verschiedenen Systeme untereinander zu realisieren, gleichzeitig werden an die Kommunikation hohe Anforderungen gestellt. Die Kommunikation zwischen den einzelnen Systemen soll sehr schnell, aber auch extrem zuverlässig funktionieren. Dies ist vor allem bei sicherheitskritischen Systemen (ABS, ESP, Airbag,...) unabdingbar.

Um eine solche anspruchsvolle Kommunikationslösung auch entsprechend kostengünstig und mit vertretbarem Aufwand realisieren zu können, werden sogenannte Bussysteme eingesetzt. Je nach Einsatzbereich eines solchen Bussystems stehen verschiedene Systeme mit unterschiedlichen Spezifikationen und Eigenschaften zur Verfügung (CAN, FlexRay, LIN, MOST). Eines jedoch eint alle diese Bussysteme: Es können Probleme und Schwierigkeiten beim Entwurf und Betrieb eines solchen Systems auftreten. Probleme bereiten vor allem die unüberschaubare Anzahl an Parametern, die den Busaufbau variieren und die Signalinte-

grität auf dem Bus nachhaltig beeinträchtigen können. Diese müssen bereits beim Entwurf des Systems berücksichtigt werden. Grundlegende Parameter können sein:

- Anzahl der Komponenten am Bus
- Terminierungen der Busleitungen
- Leitungslängen der Abzweigungen
- Bauteiltoleranzen (versch. Hersteller, Produktserien)
- Schwankungen in der Spannungsversorgung der Komponenten
- Störeinflüsse

Unter allen möglichen denkbaren Buskonfigurationen muss eine Übertragungssicherheit auf dem Bus gewährleistet sein. Für einen Busaufbau ist es daher zwingend notwendig kritische Parameterkonfigurationen zu ermitteln, unter denen eine Funktionalität des Busses nicht mehr oder nur noch eingeschränkt gewährleistet werden kann.

Diese Aufgabenstellung erfordert ein entsprechend intelligentes Herangehen. Die Möglichkeit alle erdenklichen Konfigurationen nachzubauen und messtechnisch zu untersuchen, fällt aufgrund der fast unüberschaubaren Anzahl an möglichen Konfigurationen sowie aus Zeit- und Kostengründen weg. Auf diese Weise definiert sich ein Aufgabenprofil, das typisch für den Einsatz rechnergestützter, simulationsbasierter Analysen ist.

Zu bedenken ist jedoch, dass auch bei Einsatz modernster, leistungsstarker Rechner solche simulatorischen Analysemöglichkeiten in ihrer Leistungsfähigkeit begrenzt sind.

Aufgrund der Vielzahl an Busparametern ist es somit auch beim Einsatz von Simulationen unmöglich, alle existierenden Parametervariationen zu testen. Aus diesem Grund ist es erforderlich die Analysen intelligenter und zielgerichteter durchzuführen, um so die Anzahl an benötigten Simulationen auf ein durchführbares Minimum zu reduzieren und dabei trotzdem die insgesamt kritischste Parameterkonfiguration zu ermitteln. Es bedarf also einer kompletten Analyseumgebung, die automatisiert in der Lage ist, eine solche Worstcase-Konfiguration zu berechnen.

Eine entsprechende Analyseumgebung, die diesen Anforderungen gerecht wird, ist bereits in [Gü07] entwickelt und vorgestellt worden. Diese Umgebung bildet die Grundlage der vorliegenden Arbeit. Der Aspekt der internen Steuerung der Parameterauswahl und -variation bei der Suche nach den kritischen Kombinationen wird in [Gü07] dabei nur recht knapp behandelt. Aufbauend auf den entwickelten Analysemethoden und Auswertungen beschäftigt sich die vorliegende Arbeit mit dem Thema der intelligenten Variation der Parameter, um die Analysen möglichst performant, sowohl im Hinblick auf die Laufzeit als auch hinsichtlich des Ergebnisses zu gestalten. Zu diesem Zweck werden unterschiedliche Optimierungsverfahren auf ihre Einsetzbarkeit hin untersucht und bewertet. Die Methoden, die nach einer

grundlegenden Einschätzung das größte Potential versprechen, werden implementiert und hinsichtlich ihrer Performance und Eignung für den vorliegenden Anwendungsfall untersucht und schrittweise bestmöglich an die vorliegende Problemstruktur angepasst.

## 1.2 Aufbau der Arbeit

Der folgende Abschnitt gibt einen kurzen Überblick über den Aufbau und die Struktur dieser Arbeit.

Zunächst werden einige Grundlagen und Hintergründe zu Kfz-Bussystemen und ihrer Analyse erläutert. Insbesondere wird auf die in dieser Arbeit verwendete Analyseumgebung eingegangen.

Im darauf folgenden dritten Kapitel werden unterschiedliche Optimierungsalgorithmen vorgestellt und eine geeignete Auswahl an Verfahren für eine detaillierte Analyse getroffen. Diese ausgewählten Verfahren werden im vierten Kapitel der Arbeit implementiert und in die bestehende Analyseumgebung integriert. Weiter werden zahlreiche Tests durchgeführt, um die Algorithmen möglichst optimal an die Aufgabe anzupassen und ihre Leistungsfähigkeit zu analysieren. Dazu werden unterschiedliche Aufbauten von Bussystemen genutzt, um den Einfluss der einzelnen Algorithmen-Parameter zu beurteilen und die Leistungsfähigkeit der Algorithmen zu analysieren und zu vergleichen.

Kapitel 5 demonstriert die Anwendung der Bussystemanalyse anhand eines praktischen Beispiels.

Das sechste Kapitel bildet den Abschluss der Arbeit und liefert eine Zusammenfassung der Ergebnisse.



# 2 Analyse von Kfz-Bussystemen

## 2.1 Grundlagen Kfz-Bussysteme

Die Anforderungen an heutige Kraftfahrzeuge sind außerordentlich vielseitig. Die Kunden verlangen nach immer komfortableren, sichereren und mittlerweile auch sparsameren Fahrzeugen. Um all diese Wünsche kombinieren zu können und den Anforderungen zu entsprechen, wird eine steigende Anzahl an elektronischen Systemen in die Fahrzeuge integriert. In vielen Fällen arbeiten diese Systeme aber nicht autonom für sich, sondern benötigen für ihre Arbeit zusätzliche Informationen von anderen Systemen. So ist es beispielsweise wünschenswert, dass das Radio stumm geschaltet wird wenn ein Anruf auf der eingebauten Freisprechanlage erfolgt. Dennoch sollen Anweisungen des Navigationssystems über die Bordlautsprecher auch während des Telefonats erfolgen oder die Fensterheber sollen beim Schließen der Zentralverriegelung ebenfalls automatisch angesteuert werden. Neben diesen Komfortfunktionen sind natürlich auch Sicherheitssysteme auf Informationen aus anderen Systemen angewiesen. Elektronische Stabilitätssysteme (ESC) könnten ohne zusätzliche Informationen des ABS-Systems (Raddrehzahl) ihre Arbeit nicht verrichten.

Eine Vernetzung der verbauten Systeme ermöglicht nebenbei zusätzlich eine Leistungssteigerung der einzelnen Systeme, wenn diese im Regelverbund (Global Chassis Control) mit anderen Einzelsystemen betrieben werden [WHW09]. Ein eindrucksvolles Beispiel für eine solche Leistungssteigerung liefert die sogenannte  $\mu$ -split Bremsung<sup>1</sup>. Hier konnte durch gleichzeitigen Eingriff in die Radbremse und die elektronische Servolenkung eine Verkürzung des Bremsweges um 10% erreicht werden.

Es ist offensichtlich, dass der bereits heute hohe Kommunikationsbedarf zwischen den einzelnen Systemen nicht nur erforderlich ist, sondern in Zukunft weiter stark ansteigen wird. Undenkbar ist es, die einzelnen Systeme alle einzeln untereinander zu verbinden. Hieraus würde ein unüberschaubarer Verkabelungsaufwand resultieren. Da der benötigte Platzbedarf zur Verlegung der entsprechenden Anzahl an Kabeln, sowie das durch den Einbau steigende Gewicht und die entstehenden Kosten des Fahrzeugs im Fahrzeugbau eine bedeutende Rolle spielen, wird versucht, den Verkabelungsumfang auf ein möglichst geringes Maß zu reduzieren. Aus diesem Grund werden zur Realisierung eines Kommunikationsnetz-

---

<sup>1</sup> $\mu$ -split Bremsung = Bremsung auf einer Fahrbahn mit links und rechts stark unterschiedlichen Reibwerten

werkes, über das die verschiedenen Systeme Informationen austauschen können, sogenannte Bussysteme eingesetzt.

Es stehen für die unterschiedlichen Anforderungen diverse Systeme zur Verfügung [Dor05] [Rau08] [ZS08]:

- K-Line: Der K-Line-Bus wurde entwickelt, um eine zentrale Stelle zum Auslesen der integrierten Fehlerspeicher der im Fahrzeug verbauten Steuergeräte zu schaffen. Als Kommunikationsmedium der Systeme untereinander hat er keine Bedeutung.
- CAN: Der CAN-Bus (Controller Area Network) ist das heute immer noch am meisten eingesetzte Bussystem im Kfz-Bereich. Den CAN-Bus gibt es sowohl als high-speed Variante (CAN/C, 1 Mbit/s), als auch in einer low-speed Ausführung (CAN/B, 125 kbit/s). Die schnellere Variante findet vor allem im Bereich der Motorsteuerung Verwendung. Die low-speed Version wird in erster Linie für die Steuerung von Komfortfunktionen im Fahrgastraum eingesetzt. Bei der Einsatzplanung eines CAN-Bussystems sind zwei zentrale Schwachstellen zu beachten: Aufgrund seiner maximalen Bandbreite von 1000 kBit/s ist er für den Einsatz zur Übertragung hoher Datenmengen nicht geeignet. Ebenso ist der CAN-Bus für den Einsatz in sicherheitskritischen Echtzeitsystemen, wie X-By-Wire, ungeeignet, da die Buskommunikation auf dem CSMA/CA-Verfahren<sup>2</sup> basiert.
- TT-CAN: Für den Einsatz im Rahmen von echtzeitkritischen Systemen steht mit TT-CAN (Time-Triggered-CAN) eine Weiterentwicklung des CAN-Busses zur Verfügung, die eine deterministische Datenübertragung ermöglicht. Allerdings verfügt TT-CAN über die gleiche maximale Datenrate wie der klassische CAN-Bus. Aus diesem Grund konnte sich TT-CAN nie durchsetzen, da die Datenrate für anspruchsvolle Anwendungen (X-By-Wire) weiterhin nicht ausreichend ist.
- MOST: MOST (Media Oriented Systems Transport)<sup>3</sup> ist ein Datenbus, der speziell für den Infotainmentbereich konzipiert ist [Grz08]. Die Anforderung ist daher primär eine hohe Übertragungsrate zu gewährleisten. Anforderungen bezüglich Latenzzeiten und Übertragungszuverlässigkeit spielen nur eine untergeordnete Rolle, da die übertragenden Informationen nicht für echte Steuer- oder Regelaufgaben verwendet werden [ZS08]. Als Übertragungsmedium werden Lichtwellenleiter verwendet. Dies bringt neben der großen übertragbaren Datenmenge zusätzlich den Vorteil einer

<sup>2</sup>CSMA/CA = Carrier Sense Multiple Access / Collision Avoidance: Im Falle mehrerer gleichzeitiger Sende-Anfragen erfolgt die Zuteilung der Busleitung anhand vorgegebener Objekt-Identifer, die den Inhalt einer Nachricht kennzeichnen. Über diese Identifer wird u.a. die Priorität der Nachricht festgelegt. Der Sender, der die Nachricht mit der höchsten Priorität senden will, bekommt den Bus zugeteilt. Da gerade in Übertragung befindliche Nachrichten jedoch nicht unterbrochen werden können, kann selbst für hochpriorie Nachrichten der Sendezeitpunkt nicht exakt bestimmt werden.

<sup>3</sup>MOST Cooperation: <http://www.mostcooperation.com/>

starken Unempfindlichkeit gegenüber Störungen. Nachteilig wirken sich die hohen Kosten, sowie die Verlegungsprobleme der Lichtwellenleiter aus. Ihre hohe mechanische (Knick-)Empfindlichkeit führt dazu, dass die Leitungen oftmals nicht zusammen mit den üblichen Kabelbäumen verlegt werden können, da z. B. bestimmte Mindestradien bei der Verlegung zu beachten sind. Das MOST-Protokoll unterscheidet sich grundlegend (Ringtopologie, Plug-and-Play-Funktionalität) von den Protokollen, die zur Motorsteuerung und im Fahrwerks- und Karosseriebereich eingesetzt werden [Rau08]. Die Weiterentwicklung MOST150 sieht vor allem die Steigerung der ursprünglichen 25 Mbit/s auf 150 Mbit/s vor, um den gestiegenen heutigen Anforderungen durch Videoübertragungen etc. Rechnung zu tragen.<sup>4</sup>

- D2B: Auch der D2B-Bus (Domestic Digital Bus) wird wie der MOST im Infotainmentbereich eingesetzt. Er liefert eine Bandbreite von 5,65 MBit/s und kann maximal 61 Teilnehmer adressieren. Topologisch wird eine Ringstruktur verwendet, bei der das Radio die zentrale Rolle übernimmt. Über den Bus werden sowohl Steuer- als auch Audiodaten übertragen. Wie MOST setzt D2B Lichtwellenleiter als Übertragungsmedium ein. Inzwischen ist der D2B-Bus allerdings weitgehend von den leistungstärkeren Bussen verdrängt.
- LIN: LIN (Local Interconnect Network)<sup>5</sup> erfüllt gänzlich andere Anforderungen. Mit LIN steht ein Bussystem zur Verfügung, das in erster Linie sehr kostengünstig und nur mit einem vergleichsweise geringen zusätzlichen Gewicht im Fahrzeug eingesetzt werden kann. Dies resultiert aus der Tatsache, dass beim LIN nur ein Eindraht-Übertragungsleiter verwendet wird. Verwendung findet LIN ausschließlich im Bereich von sicherheitsunkritischen Systemen, wie zum Beispiel der Ansteuerung von elektrischen Fensterhebern.
- FlexRay: FlexRay<sup>6</sup> stellt eines der neuesten Bussysteme dar. Primäre Entwicklungsziele waren zum einen eine ausreichend hohe Datenrate, mit der sich auch anspruchsvolle Steuer- und Regelungsaufgaben umsetzen lassen. Desweiteren standen die Übertragungssicherheit sowie ein deterministisches Übertragungsverhalten an vorderster Stelle, um Echtzeitanforderungen erfüllen zu können. Spezifiziert wurde FlexRay 2003 und findet inzwischen langsam steigende Verwendung im Fahrzeugbau [WH] und [HKL07].

---

<sup>4</sup>weitere Informationen zu MOST150:

<http://www.elektroniknet.de/home/automotive/technik-know-how/uebersicht/1/bussysteme/neue-uebertragungsmechanismen-fuer-most/>

<sup>5</sup>Informationen und Spezifikation unter <http://www.lin-subbus.de>

<sup>6</sup><http://www.flexray.com/>

Da das FlexRay-Bussystem Gegenstand der Untersuchungen innerhalb dieser Arbeit ist, wird es im Folgenden mit seinen einzelnen Komponenten und Bestandteilen näher vorgestellt.

### 2.1.1 Das FlexRay-Bussystem

Wie im obigen Abschnitt bereits dargestellt, versucht FlexRay die unterschiedlichen Anforderungen für sichere, aber schnelle Kommunikation mit einer hohen Übertragungsrate zu vereinen. Zu diesem Zweck definiert FlexRay ein deterministisches und fehlertolerantes Kommunikationsprotokoll, das aufgrund seiner maximalen Übertragungsrate von 10 MBit/s auch in Bereichen einsetzbar ist, in denen z. B. die Datenrate eines CAN-Busses nicht mehr ausreicht und für eine Realisierung mittels CAN mehrere Busse eingesetzt werden und aufwendig untereinander über zusätzliche Gateways verbunden werden müssten. Im Gegensatz zur ereignisgesteuerten Kommunikation eines CAN-Busses erfolgt die Kommunikation bei FlexRay zeitgesteuert. Bekannte und garantierte Zykluszeiten der Nachrichten, die nur eine sehr geringe Schwankungsbreite besitzen, machen FlexRay zu einer prädestinierten Lösung für den Einsatz in verteilten Systemen, die harten Echtzeitanforderungen unterliegen [Rau08]. Die benötigte Zeitbasis etabliert das Protokoll selbständig. Ebenso selbständig hält es diese mit einer Genauigkeit zwischen  $0,5 \mu\text{s}$  und  $10 \mu\text{s}$  synchron. Ein großer Vorteil von FlexRay ist seine Flexibilität. Da FlexRay auch zweikanalige Systeme erlaubt, können z. B. sicherheitskritische Nachrichten redundant über beide Kanäle übertragen werden, um die Verfügbarkeit eines Systems zu steigern [ZS08]. Zusätzlich ist eine Erweiterung eines FlexRay-Systems recht einfach möglich, da die Software in den vorhandenen Knoten nicht angepasst werden muss [Rau08]. Topologisch unterstützt FlexRay sowohl Bus- als auch Sternstrukturen.

#### 2.1.1.1 Komponenten des Physical Layer

Ein Bussystem stellt ein Kommunikationsmedium für die angeschlossenen Steuergeräte zur Verfügung. Folglich besteht ein Bussystem physikalisch aus Sende-, Empfangs- und Übertragungskomponenten. Abbildung 2.1 zeigt die an der physikalischen Bitübertragungsschicht, dem sogenannten Physical Layer, des FlexRay-Bussystems beteiligten Komponenten.

**Bustreiber (engl.: Transceiver)** Der Bustreiber (engl. Transceiver) ist die Schnittstelle zwischen den digitalen Signalen innerhalb der Steuergeräte und den analogen Signalen auf dem Datenbus. Jedes Steuergerät benötigt einen eigenen Transceiver. Dieser ist dafür verantwortlich, in der Sendephase die digitalen Signale der Steuergeräte in analoge Signale umzuwandeln und sie dann in den Bus einzuspeisen. Entsprechend wandelt er in der Emp-

fangsphase die auf dem Bus ankommenden analogen Signale wieder in digitale Form um und reicht sie an das jeweils angeschlossene Steuergerät weiter. Beim FlexRay-System sind die Buspegel fest spezifiziert. Gemäß der Spezifikation setzt FlexRay zur Signalübertragung einen Zweidraht-Übertragungsleiter ein. Dies bringt den Vorteil, dass sich die Informationen nicht aus dem Potential eines einzelnen Leiters gegenüber Masse ergeben, sondern aus der Potentialdifferenz der beiden Leiter. Die Signale auf Basis eines solchen differentiellen Busses sind deutlich unempfindlicher gegenüber einwirkenden Störungen als ein Einzelleiter, da sich Störungen meist identisch auf beide Leiter auswirken. Die FlexRay-Spezifikation gibt ein Potential für jeden Einzelleiter von 2,5 V vor [Fle06b]. Die entsprechende Kodierung eines Signalbits wird erreicht, indem nun die beiden Potentiale der Einzelleiter gegenüber Masse jeweils angehoben oder abgesenkt werden. Auf diese Weise entsteht beim Senden eines 1-Bits eine positive Potentialdifferenz und beim Senden eines 0-Bits eine negative Differenz zwischen den Potentialen (siehe Abbildung 2.2).

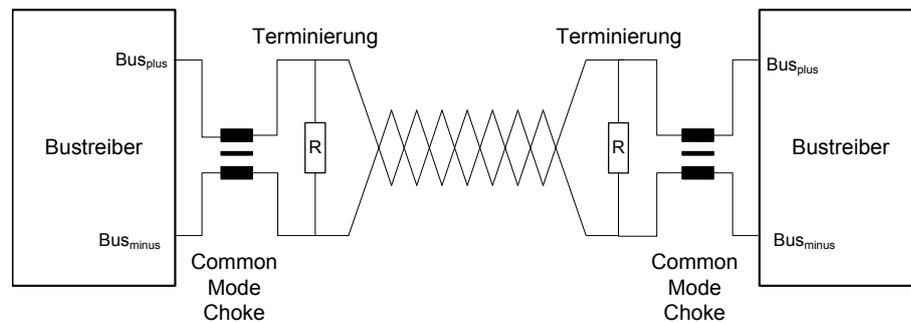


Abbildung 2.1: Komponenten des FlexRay Physical Layer

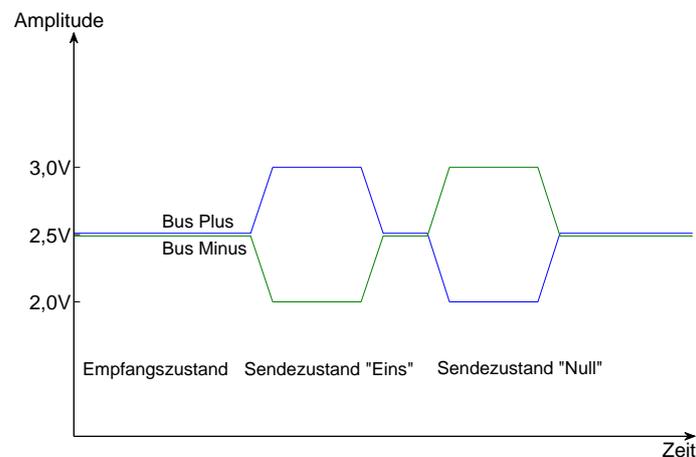


Abbildung 2.2: Signalpegel FlexRay-Bussystem

In die meisten Transceiver wird zusätzlich noch ein sogenanntes Stütznetzwerk integriert, dessen Aufgabe es ist, die Signalpegel der Einzelleiter gegenüber der Masse stabil zu halten [DO05].

**Busleitung** Als Übertragungsmedium wird für FlexRay-Bussysteme eine Leitung mit zwei gegeneinander verdrehten Adern (*Twisted-Pair*) benutzt (siehe [Fle06b]). Als gemeinsame Masseleitung fungiert die Karosserie des Fahrzeuges. Die Daten werden durch die Potentialdifferenz der beiden Leiter kodiert. So wird eine geringere Störanfälligkeit des Differenzsignals gegenüber von außen einwirkenden Störeinflüssen erreicht, da diese sich nahezu gleich auf beide Leiter auswirken. Die Verdrehung der Leiter fördert dieses Verhalten zusätzlich.

**Terminierung** Um ungewollte Reflexionen der übertragenen Signale und deren negative Einflüsse auf die Signalintegrität möglichst zu minimieren, werden die Leitungsenden durch eine Widerstandsbeschaltung abgeschlossen. Dies wird als *Terminierung* bezeichnet. Die einfachste Terminierung wird durch einen Widerstand zwischen den beiden differentiellen Busleitungen erreicht (siehe Abbildung 2.1). Besser ist es diese Terminierung in Form einer sogenannten Split-Terminierung zu realisieren, um ein besseres EMV-Verhalten<sup>7</sup> zu erreichen [Fle06a]. Dabei wird der Widerstand  $R$  in der Abbildung 2.1 durch zwei gleich große Widerstände  $R1$  und  $R2$  ersetzt. Am Mittelabgriff dieser Reihenschaltung erfolgt der Anschluss einer zusätzlichen Widerstands-Kondensator-Schaltung (RC-Glied). Um der FlexRay-Spezifikation [Fle06b] zu entsprechen, muss darauf geachtet werden, dass im Falle einer niederohmigen Terminierung die Gesamt-Buslast, die ein Teilnehmer zu treiben hat, nicht unter  $40\ \Omega$  fällt. Es können daher nicht alle Knoten korrekt im Hinblick auf den Wellenwiderstand terminiert und somit nicht alle Reflexionen verhindert werden. Üblich ist eine niederohmige Terminierung an den jeweils am weitesten voneinander entfernten Knoten, die übrigen Knoten werden hochohmig ( $1,3\ \text{k}\Omega$ ) abgeschlossen.

**Entstördrossel (engl.: Common Mode Choke)** Für eine weitere Verbesserung des EMV-Verhaltens kann zusätzlich eine *Entstördrossel* (engl.: Common Mode Choke) verwendet werden.

Bei einer Entstördrossel handelt es sich um ein passives Bauteil, mit dem der für die elektromagnetische Verträglichkeit ungünstige Gleichtaktbetrieb gehemmt werden soll. Eine solche Drossel besteht prinzipiell aus zwei Wicklungen, die bei Gleichtaktbetrieb gegenläufig vom Strom durchlaufen werden. Die induzierten Magnetfelder heben sich dabei gegenseitig auf. Somit kann eine Unterdrückung von Störemissionen beziehungsweise allgemein eine höhere Störfestigkeit erzielt werden [Rau08].

---

<sup>7</sup>EMV = elektromagnetische Verträglichkeit

Aktive Topologie-Elemente	Passive Topologie-Elemente
Knoten	Punkt-zu-Punkt Verbindung
Aktiver Stern (engl. Active Star)	passiver Stern (engl. passive Star)
	Bus

**Tabelle 2.1:** Elemente zum Aufbau eines FlexRay Systems

### 2.1.1.2 Busaufbau, Vernetzung und Topologien

Für den strukturellen Aufbau eines FlexRay-Bussystemes gibt es verschiedene Möglichkeiten die Komponenten untereinander zu verschalten. Im Folgenden werden einige wenige mögliche Busaufbauten beispielhaft gezeigt und die grundlegenden Komponenten, mit denen eine solche Vernetzung realisiert wird, erläutert [Rau08].

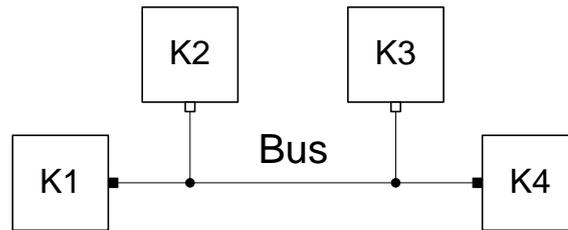
Die Tabelle 2.1 gibt einen Überblick über die unterschiedlichen Bauelemente, die zur Verfügung stehen, um ein FlexRay-Bussystem aufzubauen: Es wird zwischen aktiven und passiven Elementen unterschieden. Passive Elemente sind die *Punkt-zu-Punkt Verbindung* (Abbildung 2.3), der *passive Stern* (engl.: passive Star) (Abbildung 2.5) und der *Bus* (Abbildung 2.4). Aktive Elemente sind die *Knoten* und der *aktive Stern* (engl.: Active Star). Beim Aufbau eines FlexRay-Bussystems sind folgende Parameter zu beachten:

- Die Gesamtlänge des Systems darf bei ausschließlicher Verwendung von passiven Elementen 24 m nicht übersteigen. Beim Einsatz eines aktiven Sterns kann die Gesamtlänge auf 48 m erweitert werden, da der aktive Stern hier als birektionaler Transceiver und Repeater fungiert. Er entkoppelt die einzelnen Leitungen und verteilt die Nachrichten an die angeschlossenen Steuergeräte weiter. Bei Verwendung von zwei aktiven Sternern, kann die Buslänge auf 72 m ausgebaut werden, da auch die Verbindungsleitung zwischen beiden Sternern eine Länge von 24 m aufweisen darf.
- Es können maximal 22 aktive Komponenten an einen passiven Stern oder einen Bus angeschlossen werden.

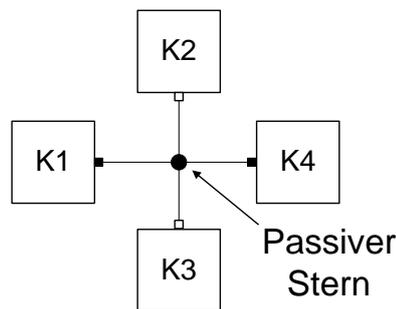
Ein Beachten dieser Reglementierungen liefert jedoch keine Funktionsgarantie für Netzwerke mit  $n \leq 22$  Komponenten und Leitungslängen unterhalb der Maximalgrenze. Die Funktionalität hängt weiterhin von den elektrischen Eigenschaften der Bustreiber in Verbindung mit den eingesetzten Terminierungen und Leitungslängen ab [Rau08].



**Abbildung 2.3:** Punkt-zu-Punkt-Verbindung zweier Knoten



**Abbildung 2.4:** Verbindung von vier Knoten über einen passiven Bus



**Abbildung 2.5:** Verbindung zwischen vier Knoten mittels passivem Stern

## 2.2 Analyseumgebung

Für die ausgiebige Analyse eines Bussystemes bedarf es leistungsstarker und automatisierter Werkzeuge. Ein solches wurde bereits in [Gü07] entwickelt und vorgestellt. Diese Analyseumgebung wird auch in dieser Arbeit verwendet und im Folgenden in ihren grundlegenden Funktionen kurz erläutert.

Die eingesetzte Analyseumgebung basiert auf der Simulation eines gegebenen Bussystems. Zunächst wird auf die Funktionen, die eine Durchführung einer solchen Simulation ermöglichen, eingegangen. Anschließend werden die Methoden und Abläufe zur Analyse der Signalintegrität eines Bussystems unter Berücksichtigung variabler Parameter erläutert.

Zur automatisierten Durchführung einer einzelnen Simulation sind folgende Arbeitsschritte in der angegebenen Reihenfolge zu erledigen:

1. Bildung eines simulationsfähigen Gesamtmodells des zu analysierenden Bussystems
2. Einsetzen der Werte der Modellparameter
3. Durchführung der Simulation

4. Einlesen der erzeugten Ergebnisdatei
5. Auswerten und Analysieren der Ergebnisse und Berechnung einer Bewertung für die aktuelle Konfiguration

Dieser komplette Ablauf wurde automatisiert. Eine grafische Darstellung dieses Ablaufs zeigt die Abbildung 2.6.

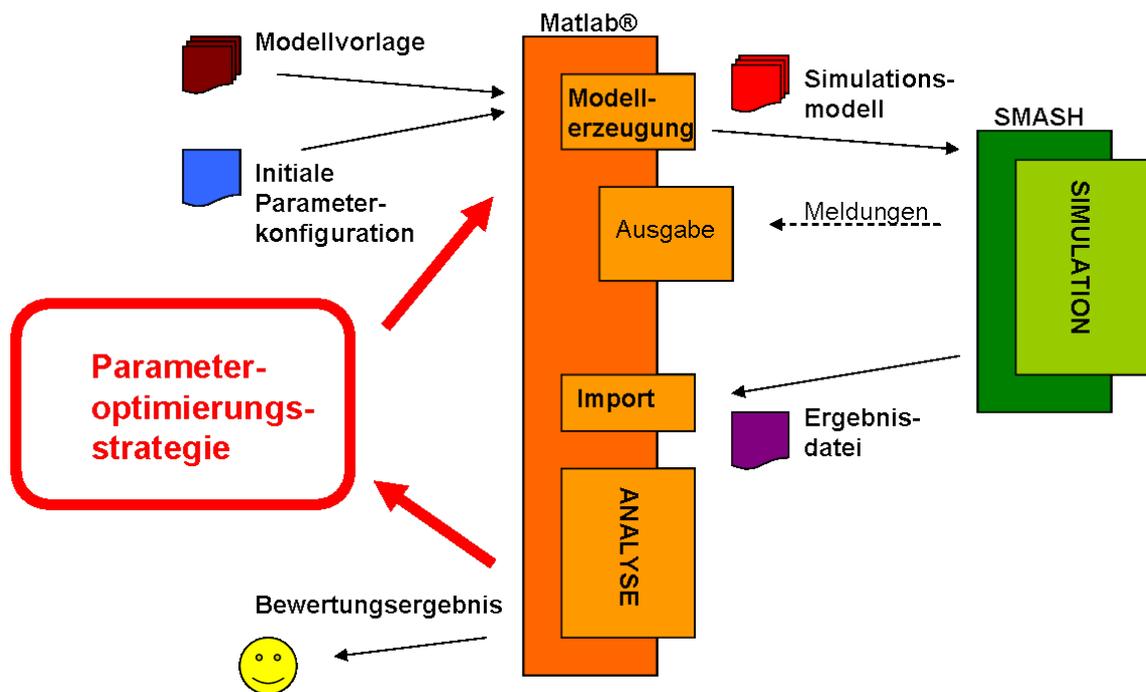


Abbildung 2.6: Analyseumgebung [Gü07]

### 2.2.1 Bildung des Simulationsmodells

Der erste Arbeitsschritt dient zur Generierung eines Gesamtmodells für die zu untersuchende Bustopologie. Die dazu benötigten Informationen werden in Form unterschiedlicher Vorlagen für die einzelnen Komponenten des Bussystem zur Verfügung gestellt. Die Informationen über den expliziten Aufbau der zu analysierenden Bustopologie müssen zusätzlich in einer Strukturdatei hinterlegt werden. Aus diesen Dateien wird dann automatisch eine Vorlage für das Simulationsmodell erzeugt.

### 2.2.2 Zuweisen der Parameterwerte

Der zweite grundlegende Arbeitsschritt, der vor dem Ausführen der eigentlichen Simulation durchgeführt werden muss, ist das Einsetzen konkreter Werte für die jeweiligen Parameter

innerhalb des Modells. Die entsprechenden Informationen sind in einer Konfigurationsdatei hinterlegt. In ihr werden für jeden Parameter des Busaufbaus sowohl sein Initialisierungswert als auch für die spätere Optimierung seine zulässigen Wertebereichsgrenzen (Minimum, Maximum) spezifiziert. Parameter, deren minimale und maximale Wertebereichsgrenze identisch sind, werden als konstant erkannt.

### 2.2.3 Simulation des Modells

Das fertige Simulationsmodell wird an einen externen VHDL-AMS Simulator übergeben. Dieser führt eine Zeitbereichsanalyse durch. Das Ergebnis dieser Analyse ist ein zeitdiskreter Signalverlauf für jeden elektrischen Knotenpunkt. Für die elektrischen Knotenpunkte, deren Signale später analysiert werden sollen, werden die Werteverläufe protokolliert und am Ende zusammen mit einer namentlich eindeutigen Kennzeichnung in einer Ergebnisdatei gespeichert. Die Spezifikation dieser Knoten wird dem Simulationstool zusammen mit dem Gesamtmodell übergeben. Typische solche Punkte sind die Busdifferenzsignale am Ende der Zweige, die Differenzsignale an den Verzweigungsknoten sowie das Differenzsignal der Signalquelle im sendenden Transceiver.

### 2.2.4 Analyse der Signale

Nach Abschluss der Simulation liegen die Signalverläufe an den spezifizierten Punkten in zeitdiskreter und wertekontinuierlicher Form gespeichert in der Ergebnisdatei vor. Bevor diese Ergebnisse nun nachfolgend analysiert werden können, müssen sie erst wieder in die Analyseumgebung importiert werden. Auch dieser Schritt geschieht automatisiert. Die namentliche Kennzeichnung der Signalverläufe, die zusammen mit den Signalwerten in der Ergebnisdatei abgelegt wird, ermöglicht dabei die Identifizierung und Zuordnung der einzelnen Signale.

Für die Bewertung der Signalintegrität sind insbesondere die Differenzsignale, die bei den einzelnen Transceivern ankommen, von Bedeutung, da diese für die Auswertung der übertragenen Nachricht maßgeblich sind. Aus diesem Grund bilden sie den zentralen Ansatzpunkt zur Analyse der Signalintegrität des zu betrachtenden Bussystems.

Die in [Gü07] entwickelte und eingesetzte Methode sieht mehrere Schritte zur Berechnung einer Bewertungszahl für den Gesamtbus vor.

Zunächst werden die aus den Simulationsergebnissen eingelesenen Signale einer Bereichsunterteilung unterzogen. Ziel dieser Einteilung ist es, einzelne Bereiche des empfangenen Signals zu identifizieren. Das Signal wird dabei in folgende Abschnitte eingeteilt: Nullpegel, ansteigende oder fallende Flanke, positiver oder negativer Signalpegel sowie Überschwingungen. Die Einteilung erfolgt auf Basis der Signalamplitude.

Der zweite vorbereitende Schritt sieht die Ermittlung eines angepassten Referenzsignals vor. Dieses dient zur Berechnung der Abweichungen zwischen Soll- und Ist-Werten. Als Aus-

gangsbasis wird das vom sendenden Transceiver auf dem Bus eingeprägte Signal verwendet, das wie folgt angepasst wird: Um die Laufzeit des Signals auf dem Bus zu berücksichtigen, muss eine zeitliche Verschiebung des Signals ermittelt werden. Das Sendesignal wird zeitlich so weit verschoben, dass der Beginn seiner ersten Flanke zeitlich mit der ersten Flanke des empfangenen Signals am betrachteten Transceiver übereinstimmt. Zusätzlich ist eine Anpassung der Amplitude des Referenzsignals notwendig, um durch Bauteile und Spannungsteiler resultierende Spannungsverluste an den Leitungsknoten zu berücksichtigen. Zu diesem Zweck wird zunächst die durchschnittliche Amplitude des zu analysierenden Signals in den ermittelten Signalpegel-Bereichen bestimmt. Nachfolgend wird das Referenzsignal in den entsprechenden Bereichen auf diesen ermittelten Durchschnittswert angepasst. Durch die beschriebenen Anpassungen wird erreicht, dass Ergebnisse aus unterschiedlichen Bus-Topologien vergleichbar werden. Ohne die erfolgten Anpassungen wären Ergebnisse aufgrund von unterschiedlichen Laufzeiten und Auswirkungen auf die Signalamplituden nur innerhalb derselben Topologie vergleichbar.

Auf Basis des angepassten Referenzsignals können nun die Abweichungen im zu analysierenden Signalverlauf ermittelt werden. Der Abweichungsfehler der Amplitude eines gemessenen Signal  $x_i$  gegenüber dem angepassten Referenzsignal  $y_i$  über insgesamt  $n$  Abtastpunkte wird als quadratischer Abweichungsfehler nach Gleichung 2.1 berechnet. Durch Einsatz einer Gewichtungsfunktion  $g_i$  wird erreicht, dass Bereiche mit einer besonderen Relevanz für die Signalintegrität - Bereiche, in denen das Signal Informationen trägt - stärker in die Berechnung der Bewertung eingehen. Hierbei finden die Ergebnisse der zu Beginn der Analyse durchgeführten Bereichsunterteilung Verwendung.

$$WMSE = \sum_{i=1}^n g_i(x_i - y_i)^2 \quad (2.1)$$

Aus den so ermittelten Bewertungszahlen für die einzelnen Signale muss nun zur abschließenden Qualitätsbewertung des Gesamtbusses ein einzelner Wert berechnet werden. Da das Ziel der Signalintegritätsanalyse das Auffinden kritischer Parameterkonfigurationen ist, wird das Maximum aller berechneten Qualitätszahlen, der an den einzelnen Transceivern ankommenden Signale, als Maß für die Signalintegrität des gesamten Bussystems verwendet.

## 2.2.5 Parametervariation

Der bisher dargelegte Ablauf führt zu einer Bewertung einer einzelnen, initialen Konfiguration des zu analysierenden Busaufbaus. Das Ziel von Analysen zur Signalintegrität eines Bussystems ist es jedoch Parameterkonfigurationen aufzufinden, die zu einer Beeinträchtigung der Signalintegrität führen. Das bedeutet, die Optimierung sucht nach den Werten der einzelnen Parameter, die in ihrer Kombination zum schlechtesten Fall für die Signal-

integrität des Busses führen beziehungsweise für die die Analyse den größten Fehlerwert ergibt.

Um diesen speziellen Fall zu bestimmen, müssen nach Abschluss der ersten initialen Simulation, basierend auf einer geeigneten Strategie, die einzelnen Parameter des gegebenen Bussystems variiert und als neue Konfiguration erneut simuliert und analysiert werden (siehe Abbildung 2.6). Dabei ist das Ziel es eine Wertekonfiguration zu bestimmen, die zu einer Verschlechterung der Signalintegrität führt. Folgende Anforderungen werden an eine solche Strategie gestellt:

- es sind möglichst wenig Simulationen zum Auffinden der kritischsten Parameterkonfiguration nötig
- der erreichte Fehlerwert, den die Strategie findet, ist möglichst maximal
- die Schwankungen im gefundenen Fehlerwert bei mehreren Durchläufen der Optimierung sind möglichst gering

Die auf diese Weise durchgeführte Suche nach schlechteren Konfigurationen wird so lange iterativ fortgesetzt bis entweder über eine bestimmte Anzahl an Variationen keine Verschlechterung der Signalqualität mehr gefunden wurde und angenommen werden kann den schlechtesten Fall gefunden zu haben, oder bis zu einer zuvor festgelegten Anzahl an maximal möglichen Iterationsschritten.

Die automatisierte Parametervariation bei der Analyse der Signalintegrität eines Bussystems ist der Kernaspekt dieser Diplomarbeit. In Kapitel 3 wird ein Überblick über verschiedene Verfahren und Strategien zur Steuerung der Parametervariation gegeben, ihre Arbeits- und Vorgehensweise erläutert und die grundlegende Eignung für den Einsatz innerhalb der Bussystemanalyse betrachtet. Im Folgenden werden besonders geeignete Verfahren ausgewählt und in die Analyseumgebung integriert. Diese werden anschließend hinsichtlich ihrer Eignung näher untersucht, speziell auf ihre Ergebnis- und Laufzeitperformance.

# 3 Optimierungsalgorithmen

Es existiert eine große Zahl an mathematischen und wissenschaftlichen Problemen, für die es ohne Schwierigkeiten möglich ist, auf analytischem Wege eine optimale Lösung zu finden. In vielen Fällen können aus der gegebenen Problemstruktur Bedingungen abgeleitet werden, die es ermöglichen, die optimale Lösung für das gegebene Problem zu ermitteln. Allerdings ist eine solche Lösung nicht für alle Problemstellungen zu erreichen. Für zahlreiche Probleme lässt sich aufgrund der Komplexität des Problems auf analytischem Wege keine Lösung konstruieren. Für diese Aufgaben muss entsprechend ein anderer Lösungsweg gefunden werden.

Für kleine Probleme besteht durchaus die Möglichkeit alle möglichen Lösungen zu testen und anschließend die beste Lösung auszuwählen. Jedoch existieren eine Reihe von Problemen, bei denen es weder möglich ist, eine optimale Lösung auf analytischem Wege zu bestimmen, noch die Möglichkeit alle Lösungen auszuprobieren eine echte Alternative darstellt. In vielen dieser Fälle wird auf rechnergestützte Simulationen zurückgegriffen, um mit Hilfe von Computern die verschiedenen Lösungsmöglichkeiten zu evaluieren und anschließend die beste auswählen zu können.

Zu dieser letztgenannten Klasse gehört auch die Analyse der Signalintegrität eines Bussystems unter Berücksichtigung von Parametervariationen. Schon bei einer kleinen Topologie kann die Anzahl der möglichen Kombinationen, die durch eine Variation der Bus-Parameter entstehen, so zahlreich werden, dass eine komplette messtechnische Überprüfung nicht mehr zu leisten ist. Bereits bei fünf variablen Parametern mit jeweils fünf möglichen Werten existieren insgesamt  $5^5 = 3125$  Kombinationsmöglichkeiten. Für diese relativ kleine Anzahl an möglichen Kombinationen ist eine vollständige Überprüfung aller Variationen mit Hilfe rechnergestützter Simulation durchaus noch denkbar. Doch auch moderne Rechner mit sehr großer Rechenkapazität stoßen schnell an ihre Grenzen. Wird die Zahl der Parameter auf zehn erhöht und sind weiterhin für jeden Parameter jeweils fünf Werte möglich, so ergibt sich bereits eine Anzahl von 9,76 Millionen Möglichkeiten. Selbst auf einem sehr schnellen Rechner würde eine Auswertung all dieser möglichen Lösungen mehrere Wochen dauern [Gü07]. Existieren gar wertekontinuierliche Parameter innerhalb des Optimierungsproblems resultiert daraus eine unendliche Anzahl an Kombinationsmöglichkeiten. Somit scheidet in diesem Fall eine komplette Bewertung aller möglichen Lösungen von vornherein

aus.

Für die Suche nach der optimalen Lösung eines solchen Problems bedarf es daher effizienter Verfahren und Methoden. Diese Verfahren müssen in der Lage sein durch ein intelligentes Vorgehen innerhalb einer angemessenen Zeit eine Lösung zu finden, die möglichst wenig von der optimalen Lösung des Problems abweicht. Solche Verfahren werden als *Optimierungsverfahren* oder *Optimierungsmethoden* bezeichnet.

Neben einer reinen Zufallssuche gibt es zwei grundlegende Ansätze, die bei der Suche nach einer optimalen Lösung angewendet werden können [GKK04]. Viele Verfahren basieren im Kern auf der Idee der lokalen Suche. Hierbei wird in der Nähe einer gefundenen, guten Lösung nach weiteren Verbesserungen gesucht. Auf die lokale Suche wird in den Ausführungen zu den verschiedenen Optimierungsverfahren näher eingegangen. Eine zweite Möglichkeit besteht darin, problemspezifische Techniken zum Einsatz zu bringen, mit denen eine Lösung Schritt für Schritt konstruiert werden kann. Dies ist jedoch nicht für alle Problemstellungen möglich.

In [Weg03] werden die bekannten Verfahren im Wesentlichen in zwei Klassen unterteilt. Die erste Klasse beinhaltet die Verfahren, die extra für ein einzelnes individuelles Problem oder eine kleine Anzahl sehr stark verwandter Problemstellungen angefertigt wurden und somit für diese maßgeschneidert sind. Der Entwurf eines solchen hochspezialisierten Verfahrens ist jedoch in der Regel sehr aufwändig und langwierig. Als Folge der perfekten Anpassung an eine kleine Zahl an Problemen zeichnen die Verfahren dieser ersten Klasse im Allgemeinen Qualitäts- sowie Geschwindigkeitsvorteile gegenüber der zweiten Verfahrensklasse aus. Diese zweite Klasse beinhaltet allgemeinere Verfahren, die mit relativ wenig Anpassungsaufwand auf eine Vielzahl unterschiedlicher Probleme angewendet werden können. In vielen praktischen Anwendungen wird daher auf diese als Heuristiken bezeichneten Verfahren zurückgegriffen, da man oftmals weder über das benötigte Problemwissen (vor allem bei neuen Problemstellungen) noch über ausreichend Zeit verfügt, die für die Entwicklung eines speziellen Verfahrens von Nöten ist.

Für den Einsatz innerhalb der Bussystemanalyse bieten sich einige dieser Heuristiken ebenfalls an.

In den folgenden Abschnitten dieses Kapitels werden zunächst ein paar Grundbegriffe der Optimierungsverfahren (Abschnitt 3.1) erläutert. Anschließend werden unterschiedliche Verfahren (Abschnitte 3.2 - 3.7) vorgestellt und hinsichtlich ihrer grundlegenden Eignung für die Anwendung im Rahmen der Kfz-Bussystemanalyse bewertet 3.8.

---

## 3.1 Grundlagen

### 3.1.1 Optimierungsprobleme

Als Optimierungsproblem werden alle solche Probleme bezeichnet, die in ihrer Problemdefinition einen oder mehrere variable Parameter beinhalten, deren Wert es zu bestimmen gilt. Das Ziel einer Optimierung ist nun die Kombination der Parameterwerte zu bestimmen, für die die zugehörige Lösung, je nach Problemdefinition, einen minimalen oder maximalen Wert annimmt.

### 3.1.2 Suchraum

Für die Formulierung eines Optimierungsproblems muss zunächst ein Suchraum  $S$  definiert werden. Dieser spezifiziert die Menge aller Lösungen, die für das Problem existieren. Der Suchraum kann durch zusätzliche Bedingungen eingeschränkt werden. So ist es durchaus üblich, dass die Werte, die ein Parameter einnehmen darf, auf einen bestimmten Wertebereich beschränkt sind. Somit werden alle möglichen Lösungen, für die der Wert des Parameters außerhalb des vorgegebenen Wertebereichs liegt, ungültig und befinden sich folglich außerhalb des Suchraums. Bestünde das Optimierungsproblem in der Anpassung zweier reeller Größen, so entspräche der Suchraum  $S = \mathbb{R}^2$ .

Der Suchraum beinhaltet also alle Wertekombinationen der Parameter, die eine gültige Lösung des Problems unter Berücksichtigung aller Einschränkungen repräsentieren. Als Folge daraus ergibt sich, dass die Größe des Suchraumes direkt von der Anzahl der Parameter und der Anzahl der jeweils möglichen Werte des Parameters abhängt und exponentiell mit dieser steigt. Für  $k$  Parameter mit jeweils  $n$  möglichen Werten, existieren genau  $n^k$  mögliche Kombinationen und somit auch  $n^k$  mögliche Lösungen des Optimierungsproblems.

### 3.1.3 Zielfunktion

Da der Suchraum lediglich die Menge aller Lösungen des Optimierungsproblems angibt, wird weiterhin eine Zielfunktion benötigt, bezüglich welcher optimiert werden soll. Der Zweck einer Optimierung besteht darin, zu einem gegebenen Problem die beste Lösung zu finden. Es muss also eine Funktion  $f: S \rightarrow \mathbb{R}$  definiert werden, die jedem Element des Suchraumes eine Bewertung zuordnet. Durch die Gestalt der Bewertungsfunktion ergibt sich eine Klassifizierung für das Optimierungsproblem als Maximierungs- oder Minimierungsproblem. Je nachdem ob z.B. die Güte der Lösung ermittelt oder der Fehlerwert einer Lösung bestimmt werden soll. Lösungstechnisch besteht kein Unterschied zwischen Maximierungs- und Minimierungsproblemen. Jedes Minimierungsproblem  $f$  kann durch ein Maximierungsproblem  $f' = -f$  berechnet werden [Kin94].

In dem oben genannten Beispiel eines zweidimensionalen, reellen Suchraums beschreibt die

Zielfunktion eine Funktion  $g: \mathbb{R}^2 \rightarrow \mathbb{R}$ . Bei einem Maximierungsproblem wird die Kombination der beiden reellwertigen Parameter der Funktion  $g$  gesucht für die die Funktion  $g$  den maximalen Wert annimmt.

## 3.2 Hill-Climbing

Die Optimierungsmethode basierend auf dem Hill-Climbing-Algorithmus stellt eine der einfachsten Strategien zur Optimierung von Lösungen dar. Die Grundidee ist es, ausgehend von einer zufällig gewählten Lösung zu starten und dann einen wiederum zufälligen Schritt zu einer benachbarten Lösung zu vollziehen. Ist die Bewertung der Nachbarlösung besser als die vorherige, so wird von dieser Lösung aus weitergesucht, andernfalls wird in der Nachbarschaft der alten Lösung weitergesucht. Die Definition einer geeigneten Nachbarschaft ist stark abhängig von der jeweiligen Problemstruktur.

Soll die Zielfunktion  $f: S \rightarrow \mathbb{R}$  maximiert werden, läuft der Algorithmus wie folgt ab [GKK04]:

1. Wähle einen Startpunkt  $s_0 \in S$ .
2. Wähle  $s_1 \in S$  in der Nähe von  $s_0$ .
3. Falls  $f(s_1) \geq f(s_0)$ , wähle  $s_1$  als neuen Startpunkt und fahre mit 2. fort; ansonsten weiter mit  $s_0$  als Startpunkt und wähle ein neues  $s_1$  in Schritt 2.

Die vergleichsweise einfache Struktur des Algorithmus führt zu einer Reihe von Problemen, auf die bei einer Anwendung zu achten ist [GKK04].

- Der Algorithmus ist nicht in der Lage ein lokales Optimum wieder zu verlassen, nachdem er es gefunden hat.
- Die Wahrscheinlichkeit, dass der Algorithmus ausgehend von dem zufällig gewählten Startpunkt nur ein benachbartes lokales Optimum findet, ist sehr groß, da der Algorithmus jede Verbesserung sofort und ausnahmslos akzeptiert.
- Die Definition einer geeigneten Nachbarschaft (Schrittweite) ist nicht trivial, insbesondere ist dies im Falle von stetigen Problemstellungen schwierig.

Die Abbildung 3.1 verdeutlicht das Problem der lokalen Optima. Der Algorithmus hat von seinem Startpunkt aus nacheinander die beiden Zwischenlösungen (rot) gefunden. Bei der weiteren Suche ist der Algorithmus nur in der Lage ein lokales Optimum (türkis) zu finden. Hat er diese lokal optimale Lösung gefunden, terminiert die Suche, weil in der Nachbarschaft keine bessere Lösung mehr existiert. Die bessere Lösung (lila) sowie das globale Optimum (braun) findet der Algorithmus hingegen nicht.

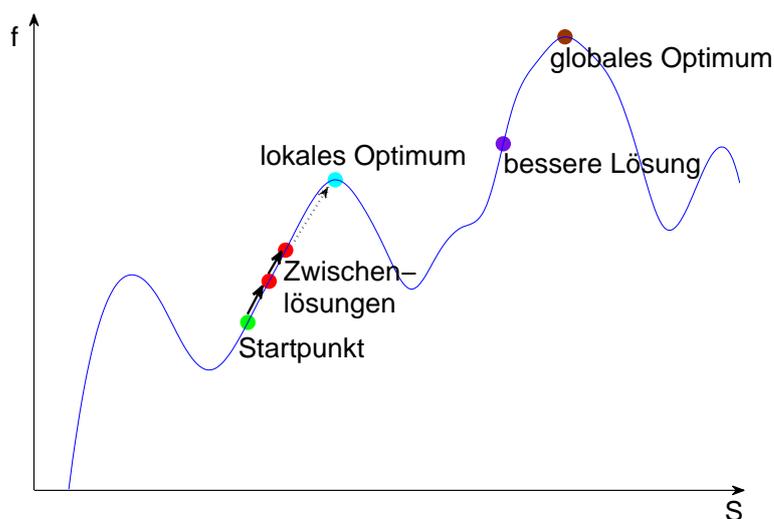


Abbildung 3.1: Hill-Climbing

Um insbesondere mit dem Problem der lokalen Optima besser umgehen zu können, existieren verschiedene Ansätze und Weiterentwicklungen des einfachen Hill-Climbing-Algorithmus. Die einfachste Methode besteht darin, eine größere Anzahl von Suchläufen zu absolvieren und die Startpunkte dabei möglichst gleichverteilt über den Suchraum zu streuen. Alternativ ist es möglich, alle benachbarten Lösungen zu betrachten und dabei die jeweils beste zu wählen. Dieser Ansatz wird als *Steepest-Ascent-Hill-Climbing* bezeichnet [Mit96]. Eine derartige Abwandlung des ursprünglichen Algorithmus bewirkt, dass sich die Suche stets in Richtung der größten Verbesserung bewegt. Dadurch wird in der Regel nur das lokale Optimum am Ende des steilsten Anstiegs in der Nähe gefunden, aber nicht das globale Optimum.

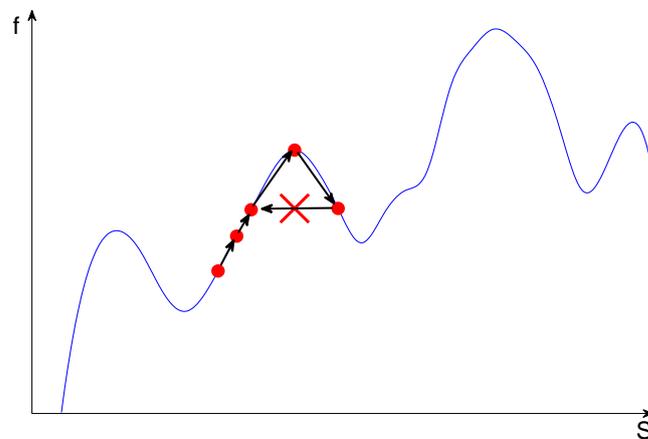
Aufgrund der beschriebenen Probleme und Schwierigkeiten kommt ein ausschließlich lokaler (greedy) Suchalgorithmus wie Hill-Climbing für einen sehr komplexen Suchraum, wie er im vorliegenden Anwendungsfall zu erwarten ist, nicht in Frage.

### 3.3 Tabu-Suche

Die Tabu-Suche ist ein bereits 1986 in [Glo86] vorgestelltes Verfahren, das den Hill-Climbing-Ansatz aufgreift und erweitert. Durch diese Erweiterung sollen die Fähigkeiten im Umgang mit lokalen Optima erweitert werden [BK05]. Insbesondere soll der Algorithmus in der Lage sein, ein lokales Optimum wieder zu verlassen, wenn er eines gefunden hat. Zu diesem Zweck werden zusätzliche Informationen über den beschrittenen Suchweg gespeichert und verwaltet. Wie beim Hill-Climbing ist die Grundlage des Suchverfahren eine Nachbar-

schaftsbeziehung zwischen den Lösungen, die beispielsweise über den Abstand zwischen den Lösungen definiert werden kann. Im Gegensatz zum Hill-Climbing-Algorithmus akzeptiert die Tabu-Suche auf ihrem Weg durch den Suchraum jedoch nicht ausschließlich nur Verbesserungen, sondern auch ggf. kleine Verschlechterungen [GKK04]. Von der Idee her sollen keine oder möglichst wenige Schritte durchgeführt werden, die die Auswirkungen vorheriger Schritte wieder rückgängig machen.

Dieses Ziel wird durch das Verwalten von zwei Listen erreicht: Die *Tabu-Liste*  $T$  und die *Running-Liste*  $R$ . Während in die Running-Liste  $R$  alle durchgeführten Schritte eingetragen werden, hat die Tabu-Liste  $T$  eine feste vordefinierte Länge  $k$  und bezieht sich lediglich auf die letzten  $k$  Schritte. In der Tabu-Liste werden die Komplemente der letzten  $k$  durchgeführten Schritte beziehungsweise die Komplemente der durchgeführten Veränderungen, der zu optimierenden Parameter gespeichert. Sie sind für den jeweils aktuell durchzuführenden Schritt tabu. Auf diese Weise kann einfach überprüft werden, ob der aktuelle Schritt einen der letzten  $k$  Schritte rückgängig machen würde. Die Überprüfung verhindert, dass die Suche einen Kreis durchläuft, wie er üblicherweise beim Auffinden eines lokalen Optimums auftritt (siehe Abbildung 3.2). Abbildung 3.3 zeigt den schematischen Ablauf einer Tabu-Suche.



**Abbildung 3.2:** Tabu-Suche

Typische Abbruchkriterien für die Tabu-Suche sind

- eine feste Anzahl an Iterationen;
- eine bestimmte Anzahl an Iterationen ohne gefundene Verbesserung;
- das Erreichen eines vorab spezifizierten Ergebnisniveaus.

Trotz der besseren Fähigkeit mit lokalen Optima umzugehen, stellt die Tabu-Suche kaum ein geeignetes Verfahren für den Einsatz innerhalb der Bussystemanalyse dar. Sie ist zwar in der Lage, ein einmal gefundenes lokales Optimum wieder zu verlassen, so wie vorgegangen wird, hat die Tabu-Suche jedoch einen entscheidenden Nachteil. Sie tendiert dazu immer noch zu sehr lokal zu operieren. Die Tabu-Suche verbringt die meiste Zeit mit der Suche in einem sehr stark begrenzten Teilbereich des Suchraumes. Als Konsequenz dieser Tatsache ergibt sich eine sehr große Wahrscheinlichkeit dafür, dass der interessanteste Teil des Suchraumes gar nicht betrachtet wird. In diesem Fall würden somit ausschließlich Ergebnisse gefunden, die immer noch sehr weit entfernt vom globalen Optimum liegen [BK05]. Einen Überblick zur Tabu-Suche liefert Glover in [Glo89, Glo90].

- 
1. Wähle einen Startpunkt  $s_0 \in S$ .
  2. Suche in der Nachbarschaft von  $s_0$  die Lösung  $s$  mit der größten Verbesserung (bzw. die Lösung mit der geringsten Verschlechterung) gegenüber  $s_0$ , die durch einen Schritt  $m$  erreicht wird, der nicht durch die Tabu-Liste ausgeschlossen ist.
  3. Füge den Schritt  $m$  in die Running-Liste ein:  $R_1 = R_0 \cup \{m\}$ .
  4. Füge die Negation  $\bar{m}$  zu  $m$  in die Tabu-Liste ein:  $T_1 = T_0 \cup \{\bar{m}\}$ .
  5. Wähle  $s$  als neuen Startpunkt für den nächsten Schritt:  $s_0 = s$ .
  6. Überprüfe die Stoppkriterien, bei Nichterfüllung weiter bei 2.
- 

**Abbildung 3.3:** schematischer Ablauf Tabu-Suche nach [GKK04]

### 3.4 Simulated Annealing

Eine Weiterentwicklung des bereits 1953 vorgestellten Metropolis-Algorithmus [MRR<sup>+</sup>53] stellt *Simulated Annealing* dar. Im Kern basiert Simulated Annealing ebenfalls auf der Idee der lokalen Suche, beherrscht jedoch deutlich besser den Umgang mit lokalen Optima als der Hill-Climbing-Algorithmus oder die Tabu-Suche. Die Inspiration zur Entwicklung des Simulated Annealing Algorithmus lieferten die physikalischen Vorgänge, die sich beispielsweise beim langsamen Abkühlen eines geschmolzenen Metalls beobachten lassen [Vöc08]. In einer Metallschmelze können sich die einzelnen Atome frei und ungehindert bewegen. Durch das Abkühlen der Schmelze wird die Bewegungsfreiheit der Atome zunehmend eingeschränkt und das Metall verfestigt sich. Dabei bildet sich eine kristalline Struktur zwischen den Atomen. Bei weiter sinkender Temperatur gehen immer mehr Teilchen in den für sie energetisch günstigeren kristallinen Zustand über. Nur sehr wenige Teilchen verlassen die-

sen Kristallverbund wieder. Mit sinkender Temperatur gegen Ende des Abkühlvorganges wird die Wahrscheinlichkeit, dass ein Atom den Kristallverbund wieder verlässt, immer geringer. Wie stark der Verbund der Atome untereinander ist, hängt nun primär von der Geschwindigkeit ab, mit der der Abkühlungsprozess abläuft. Sinkt die Temperatur zu schnell, bilden sich mehrere Kristalle gleichzeitig aus. Daraus würde letztendlich ein polykristalliner Verbund entstehen. Spezielle Anwendungen erfordern jedoch, dass lediglich ein einzelner Kristall erzeugt wird (z.B. Siliziumkristalle in der Chipindustrie). Diese monokristalline Struktur stellt das niedrigste Energieniveau dar. Beim Erstarren der Metalls wird somit versucht das globale Energieminimum entstehen zu lassen. Technisch wird dies im Falle der Siliziumkristalle erreicht, indem ein einzelner stabförmiger Kristall langsam aus der Schmelze gezogen wird. Aufgrund des daraus resultierenden sehr langsamen Abkühlungsprozesses haben die Atome ausreichend Zeit, sich richtig im Kristallverbund zu positionieren. Die Güte des entstehenden Kristalls ist also direkt abhängig von zwei Faktoren: Zum einen von der Temperatur, zum anderen von der Geschwindigkeit mit der sich die Temperatur während des Abkühlvorgangs reduziert. Der Vorgang, mit dem versucht wird das globale Energieminimum, ausgehend von einer Metallschmelze, entstehen zu lassen, lässt sich, wie im folgenden Abschnitt beschrieben, auf Optimierungsprobleme übertragen.

### 3.4.1 Der Algorithmus

Zu Beginn der Optimierung ist das Problem der Metallschmelze vergleichbar. Die Parameter, die optimiert werden sollen, werden mehr oder weniger zufällig gewählt. Ausgehend von dieser initialen Lösung  $x$  wird nun eine gewisse Anzahl an neuen Lösungen  $x'$  berechnet, die in einer (zu definierenden) Nachbarschaft der bisherigen Lösung liegen. Dabei werden alle besseren Lösungen als die aktuelle (bei Maximierungsproblemen  $x' \geq x$ ) als neue Lösung akzeptiert. Eine schlechtere Lösung als die bisherigen wird mit der Wahrscheinlichkeit

$$p = e^{\frac{\Delta E}{k_b T}} \quad (3.1)$$

akzeptiert.  $\Delta E$  bezeichnet die Energiedifferenz  $E_x - E_{x'}$  (Qualität) der beiden Lösungen,  $k_b$  ist die Boltzmann-Konstante,  $T$  die aktuelle Temperatur [BK05].

Eine Verbesserung der Lösung entspricht dem Anlagern eines Atoms an den Kristall im physikalischen Vorbild des Algorithmus, eine Verschlechterung ist gleichzusetzen mit einem Schrumpfen des Kristalls, also dem Zurückkehren einzelner Atome in die Schmelze. Zu Beginn der Optimierung ist die Temperatur hoch, Verschlechterungen werden demnach durchaus häufig akzeptiert. Im Laufe des Optimierungsvorganges wird dann der Abkühlungsprozess simuliert, indem die Temperatur schrittweise langsam abgesenkt wird. Je größer die Energiedifferenz (Differenz der Lösungsqualität) und je geringer die Temperatur werden, desto unwahrscheinlicher wird eine Änderung akzeptiert, die zu keiner Verbesserung führt. Mit fortlaufender Dauer des Optimierungsvorganges sinkt die Wahrscheinlich-

keit für die Akzeptanz einer Verschlechterung der Lösungsqualität zunehmend. Technisch ist die Umsetzung des Algorithmus relativ einfach: Ein Kontrollparameter  $c$  repräsentiert die aktuelle Temperatur  $T$ . Ausgehend von der initial berechneten Lösung wird zunächst eine feste Anzahl an Folge­lösungen berechnet. Dabei wird wie beschrieben eine Verbesserung stets akzeptiert, eine Verschlechterung wird mit einer Wahrscheinlichkeit akzeptiert, die vom aktuellen Wert des Kontrollparameters  $c$  abhängig ist. Im Anschluss wird der Wert des Kontrollparameters reduziert, was folglich auch die Akzeptanzwahrscheinlichkeit für schlechtere Lösungen verringert. Der Optimierungsvorgang terminiert, wenn der Kontrollparameter einen festgelegten Wert unterschritten hat, eine bestimmte Qualität der Lösung gefunden wurde oder wenn über eine definierte Anzahl berechneter Lösungen keine Verbesserung mehr gefunden wurde. Der Vorteil des Verfahrens gegenüber der einfachen lokalen Suche, wie sie der Hill-Climbing-Algorithmus einsetzt, liegt darin, dass einmal gefundene lokale Optima mit einer Wahrscheinlichkeit  $p > 0$  wieder verlassen werden (siehe Abbildung 3.4), insbesondere wenn es sich um schlechte lokale Optima handelt.

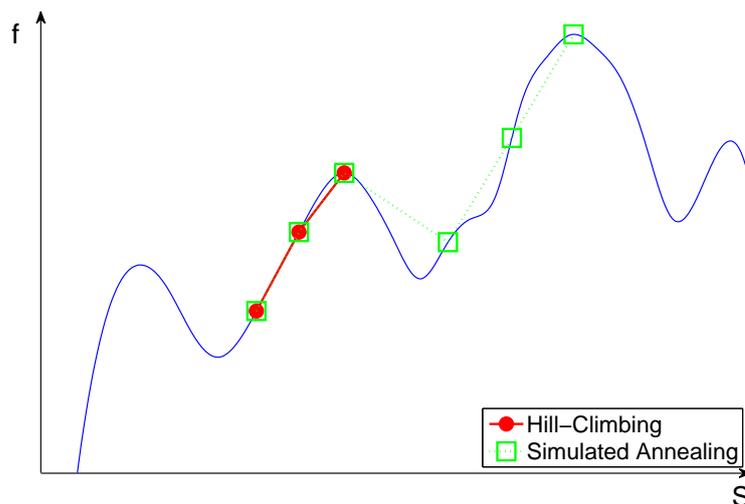


Abbildung 3.4: Simulated Annealing im Vergleich zu Hill-Climbing

### 3.4.2 Anpassungen und Erweiterungen

**Parameterwahl** Die Auswahl des Parameters, der im Vergleich zur bisherigen Lösung variiert werden soll, kann auf unterschiedliche Weise gesteuert werden. Denkbar sind sowohl eine uniform zufällige Auswahl des Parameters sowie eine festgelegte Reihenfolge anhand der die Parameter nacheinander ausgewählt.

Auch die Veränderung des Wertes des ausgewählten Parameters kann unterschiedlich realisiert werden. Bei diskreten Parametern bietet es sich beispielsweise an, den aktuellen Wert

um eine Einheit zu vergrößern oder zu verkleinern. Bei kontinuierlichen Parametern ist eine alternative Umsetzung erforderlich. Denkbar ist, den Wert des Parameters um einen festgelegten (prozentualen) Wert zu variieren.

**Lokalisierungsschritte** Lokalisierungsschritte stellen eine Technik dar, mit der die Effektivität des Standard-Algorithmus des Simulated Annealing weiter verbessert werden kann [BK05]. Lokalisierungsschritte haben zum Ziel, die Suche zunächst auf den aktuellen Parameter zu fokussieren, wenn eine Veränderung seines zugewiesenen Wertes im letzten Schritt zu einer Verbesserung geführt hat. Im nächsten Schritt wird also zunächst kein neuer Parameter ausgewählt dessen Wert variiert werden soll, sondern erneut der Wert des aktuell ausgewählten Parameters verändert. Diese Fokussierung auf einen Parameter wird solange aufrecht erhalten, bis eine Veränderung des Parameterwertes keine Verbesserung mehr erzielt.

**Cooling-Schedule** Eine weitere Möglichkeit den algorithmischen Verlaufes der Berechnungen anzupassen besteht darin, das sogenannte *Cooling-Schedule* zu verändern. Das Cooling-Schedule definiert wie der Kontrollparameter, also die „Temperatur“ im Algorithmus, im Laufe der Optimierung dekrementiert wird. Einfache Varianten sind zum Beispiel [Rud93]

- lineares Dekrementieren:  $T(i) = \frac{T_0}{i+1}$ ,  $i = 0, 1, 2, \dots$
- logarithmisches Dekrementieren:  $T(i) = \frac{T_0}{\log(i+2)}$ ,  $i = 1, 2, \dots$
- geometrisches Dekrementieren:  $T(i) = c^i \cdot T_0$ ,  $c \in (0, 1)$

### 3.5 Ameisenkolonieoptimierung

Die Idee der Ameisenkolonieoptimierung (engl.: Ant Colony Optimization, ACO), die zur Klasse der Schwarm-Algorithmen gehört, entstammt Beobachtungen von Ameisen und ihren Fähigkeiten kürzeste Wege zum Beispiel zu Nahrungsquellen zu finden (siehe [DMC96]). Durch den Algorithmus wird versucht das Verhalten der Ameisen nachzubilden und auf Optimierungsprobleme anzuwenden.

Eine Ameise hinterlässt auf ihrem Weg Pheromone. Diese Pheromone werden von anderen Ameisen wahrgenommen und beeinflussen ihre Entscheidungen bezüglich des eigenen Weges. Nimmt eine Ameise in ihrer Umgebung keine Pheromone wahr, so entspricht ihr Weg einem Zufallspfad. Findet sie jedoch in ihrer Umgebung eine Pheromonspur, so folgt sie dieser mit einer hohen Wahrscheinlichkeit. Findet die Ameise mehrere solcher Pheromonspuren, folgt sie mit der größten Wahrscheinlichkeit der stärksten Spur. Auf diese Weise, wenn auch hier sehr vereinfacht dargestellt, sind Ameisen in der Lage den kürzesten Weg zu ihrem Ziel zu finden, ohne direkt miteinander kommunizieren zu müssen. Existieren

mehrere Pfade unterschiedlicher Länge, so wird der kürzeste Pfad mit der Zeit von immer mehr Ameisen benutzt. Dies resultiert aus der Tatsache, dass die Pheromone im Laufe der Zeit kontinuierlich abgebaut werden, so dass bei gleicher Frequentierung unterschiedlich langer Pfade die Konzentration der Pheromone auf dem kürzesten Weg am höchsten ist. Die Übertragung auf Optimierungsprobleme funktioniert folgendermaßen: Die Aktionen der Ameisen werden als Konstruktion eines kürzesten Pfades (als Konstruktion einer Lösung des Optimierungsproblems) interpretiert. Diese Herangehensweise erfordert, dass sich eine gültige Lösung des Optimierungsproblems über Zwischenlösungen konstruieren lässt, denn eine Ameise konstruiert einen Pfad, indem sie an jeder Verzweigung entscheidet, welcher Pheromonspur sie folgt.

### 3.5.1 Formale Anforderungen an die Problemstruktur

Folgende Anforderungen werden an die Struktur des Optimierungsproblems gestellt um eine Optimierung mittels ACO durchführen zu können [GKK04]:

- Die einzelnen Lösungen des Problems bestehen aus Sequenzen der Form  $(n_1, \dots, n_l)$ , diese müssen nicht notwendigerweise dieselbe Länge haben.
- Es gibt eine Menge  $\Omega$  von Einschränkungen, durch die die Menge der gültigen Lösungen begrenzt wird, d.h. bestimmte Sequenzen sind verboten.
- Auf der Menge aller Sequenzen existiert eine Nachbarschaftsbeziehung. Diese lässt auch unvollständige Sequenzen zu, die nicht den Einschränkungen aus  $\Omega$  widersprechen, allerdings erst noch zu einer vollständigen Lösung ausgebaut werden müssen. Die Nachbarschaftsbeziehung beschreibt, wie von einer unvollständigen Lösung, also einer Sequenz der Länge  $n$ , zu einer Sequenz der Länge  $\geq n+1$  gelangt werden kann.
- Es ist nicht zwingend notwendig, dass sich die Zielfunktion auch auf Teilprobleme, also unvollständige Sequenzen, anwenden lässt.

### 3.5.2 Schematischer Ablauf einer Optimierung

Im Folgenden werden die grundlegenden Prinzipien der Ameisenkolonieoptimierung verdeutlicht [GKK04]:

- Es werden ständig neue Ameisen auf den Weg geschickt. Eine Ameise konstruiert schrittweise eine Lösung, indem sie ausgehend von der leeren Sequenz, diese über Nachbarschaftsbeziehungen erweitert und ausbaut, bis eine vollständige Lösung erarbeitet wurde.
- Auf ihrem Weg zu einer Lösung hinterlässt die einzelne Ameise Pheromone an den Stellen, die den Übergang zwischen den Sequenzen darstellen, d.h. an den Übergängen

zwischen zwei benachbarten Zuständen. Die Menge der hinterlassenen Pheromone wird dabei allerdings erst nach Ende der Konstruktion bestimmt, also dann wenn die Ameise eine vollständige Lösung konstruiert hat. Dieses Vorgehen unterscheidet den Algorithmus vom natürlichen Vorbild, da dort die Pheromone schon beim Benutzen des Weges hinterlassen werden. Der Algorithmus hingegen verteilt die Pheromone nach vollständiger Konstruktion des Wegs nachträglich an den Sequenzübergängen. Die Menge an Pheromonen wird in Abhängigkeit von der Güte der erzeugten Lösung bestimmt. Auch hier unterscheidet sich der Algorithmus vom Verhalten der realen Ameisen, diese hinterlassen eine konstante Menge an Pheromonen, die nicht von der Nützlichkeit des Weges beeinflusst wird.

- Bei der Konstruktion einer neuen Lösung lässt sich die Ameise von den Pheromonen leiten, die sie vorfindet. Die Ameise wählt an Entscheidungspunkten mit einer größeren Wahrscheinlichkeit eine der zur Auswahl stehenden Möglichkeiten, die mit einer vergleichsweise hohen Pheromonmenge markiert ist.
- Die an den Übergängen hinterlassenen Pheromone verflüchtigen sich im Laufe der Zeit. Nur wenn immer wieder Ameisen diesen Übergang auswählen und benutzen hält sich die Menge der Pheromone, die eine Auswahlmöglichkeit markiert, auf einem gleich bleibend hohen Niveau. Die Pheromonmenge steigt gar an, wenn immer mehr Ameisen diesen Übergang auswählen.

Ein gern zitiertes Beispiel für ein Problem, dass sich mittels ACO optimieren lässt, ist das *Handelsreisenden-Problem* (traveling salesman problem, TSP).

Hierbei wird die günstigste (kürzeste) Rundreise gesucht, die eine festgelegte Anzahl an Städten miteinander verbindet, so dass jede Stadt genau einmal besucht und wieder zum Ausgangspunkt zurückgekehrt wird. Die einzelnen Verbindungswege zwischen den Städten werden dabei entsprechend ihrer Entfernungen bewertet. Eine Gesamtbewertung ergibt sich dann aus der Summe aller Wege des Handelsreisenden auf seiner Rundreise [Weg03]. Wird das Problem mittels des ACO-Algorithmus gelöst, können die Verbindungen zwischen den Städten als Übergang zwischen den Sequenzen definiert werden. Je kürzer eine solche Verbindung ist, desto mehr Pheromone werden auf ihr hinterlassen. Auf diese Weise kann die Ameise schrittweise eine gültige Lösung konstruieren [GKK04].

Aus dem gegebenen Anwendungsbeispiel und den Anforderungen an die Problemstruktur ist erkennbar, dass sich das ACO-Verfahren besonders für Optimierungsprobleme eignet, die sich in Form von Graphen darstellen lassen. Insbesondere sind die Zustandsübergänge zwischen den Sequenzen anzugeben, die auf einer zu definierenden Nachbarschaftsbeziehung zwischen den Sequenzen basieren. Gerade diese notwendige Definition der Nachbarschaft stellt in einigen Optimierungsproblemen eine nicht triviale Aufgabe dar. Existieren

---

Parameter innerhalb des Optimierungsproblems, die kontinuierliche Werte annehmen können, wird die Erstellung einer solchen Definition zusätzlich erschwert.

## 3.6 Genetische Algorithmen

Als Vorbild für die Entwicklung der genetischen Algorithmen dient die biologische Genetik, insbesondere die Evolutionstheorie. Die heutige synthetische Evolutionstheorie gründet sich auf drei zentrale Bestandteile: Die Vorstellung Charles Darwins eines stetigen Wechselspiels zwischen Selektion und Variation, die klassische Genetik sowie die moderne Populationsgenetik.

### 3.6.1 Grundlagen

Bereits 1859 veröffentlichte Charles Darwin in “On the Origin of Species by Means of Natural Selection“ seine Evolutionstheorie und erklärt die Evolution als einen stufenweisen Prozess [Dar63]. Zentraler Aspekt seiner Ausführungen ist ein stetiges Wechselspiel zwischen Mutation und Selektion. Weiterhin legte Darwin folgende Sachverhalte dar:

- Lebewesen erzeugen mehr Nachkommen als zum Erhalt der Rasse nötig wären. Nicht alle dieser Nachkommen können überleben.
- Die einzelnen Nachkommen eines Elternpaares zeigen unterschiedliche Eigenschaften und Fähigkeiten.
- Die Lebewesen, deren Eigenschaften zur Nahrungsgewinnung, zur Ausnutzung des vorhandenen Lebensraumes und der Fortpflanzung am besten an die Umgebung angepasst sind, überleben und verdrängen die anderen.

Implizit setzt die darwinistische Evolutionstheorie somit folgende Aspekte als gegeben voraus [Kin94]:

- Es existiert eine Population an Individuen.
- Die Individuen innerhalb einer Population unterscheiden sich in ihren definierenden Eigenschaften und Fähigkeiten.
- Die Überlebensfähigkeit (Fitness) eines Individuums hängt maßgeblich von seinen Eigenschaften ab.
- Die Individuen der Population besitzen die Fähigkeit zur Reproduktion.

Zu beachten ist dabei, dass die Fitness eines Individuums nicht nur die eigene Überlebensfähigkeit kennzeichnet, sondern sich insbesondere auch auf den Erfolg hinsichtlich der

Zeugung von Nachkommen im Vergleich zu anderen Individuen bezieht. Die Individuen, die aufgrund ihrer Eigenschaften besser an die Umweltbedingungen angepasst sind, haben eine höhere Chance Nachkommen zu zeugen und so ihre Gene weiterzugeben. Dies alles führt zu einem natürlichen Ausleseprozess, in dem sich die am besten an die Umwelt angepassten Individuen durchsetzen (“Survival of the fittest“) und so über Generationen zu einer allmählichen Veränderung der Arten führen.

Die Evolution ist demnach in der Lage, über die Zeit hinweg Lebewesen hervorzubringen, die optimal an die Bedingungen ihres Lebensraumes angepasst sind. Diese Tatsache stellt den Ausgangspunkt zu einer neuen Klasse von Such- und Optimierungsverfahren dar, den *Evolutionären Algorithmen*.

Evolutionäre Algorithmen versuchen den Ablauf der Evolution auf einer abstrakten Ebene zu imitieren, um so vielfältige und komplexe Problemstellungen effizienter bearbeiten zu können. Die verbreitetste Form der Evolutionären Algorithmen sind die *genetischen Algorithmen*. Ihre Entwicklung geht zurück auf die 1975 von John Holland vorgestellten Ideen [Hol92].

### 3.6.2 Terminologie

Für das Verständnis von genetischen Algorithmen ist es zunächst sinnvoll, einen Blick auf die verwendete Terminologie zu richten. Vielfach leiten sich die einzelnen Begriffe von den ursprünglichen biologischen Vorbildern ab, bezeichnen in der Regel im algorithmischen Gebrauch aber einen eher abstrakten Sachverhalt [GKK04].

Eine einzelne Lösung des gegebenen Problems wird gemeinhin als *Individuum* oder als *Chromosom* bezeichnet. Diese beiden Ausdrücke werden in der Literatur weitgehend synonym verwendet. In dieser Arbeit wird fortan der Begriff Individuum verwendet. Äquivalent zum natürlichen Vorbild verfügt jedes Individuum über eine Überlebensfähigkeit, diese bezeichnet seine Eignung für die vorliegende Aufgabe. Die Überlebensfähigkeit wird als *Fitness* bezeichnet. Entsprechend zur Evolutionstheorie verfügt das Individuum mit der größten Fitness über die beste Anpassung an seinen Lebensraum und repräsentiert somit die optimale Lösung des gegebenen Problems. Ziel der genetischen Algorithmen ist es nun genau dieses Individuum zu finden beziehungsweise durch Imitieren von evolutionstheoretischen Abläufen zu generieren.

Mehrere Individuen werden zu einer *Population* zusammengefasst und bilden dann die jeweils aktuelle Lösungsmenge des genetischen Algorithmus. Durch das Nachbilden der evolutionären Abläufe werden Populationen im Laufe der Zeit durch neue Populationen ersetzt. Diese Iterationsstufen werden als *Generationen* bezeichnet.

Jedes einzelne Individuum besitzt eine Reihe von Eigenschaften, die maßgeblich für seine Fitness sind. Diese Eigenschaften werden als *Gene* bezeichnet. Die Gene repräsentieren die Parameter des gegebenen Optimierungsproblems. Ihre jeweilige Ausprägung, die als *Allel*

bezeichnet wird, beschreibt letztendlich die einzelne Lösung, die das jeweilige Individuum repräsentiert.

Entsprechend dem Vorbild der Evolution müssen aus den aktuell vorhandenen Individuen auf geeignete Weise neue Individuen erzeugt werden. Demnach müssen die Individuen in der Lage sein sich zu reproduzieren. Für diesen Zweck stehen sogenannte Crossover- und Mutationsoperatoren zur Verfügung. Diese haben die Aufgabe, die Weitergabe von Eigenschaften bei der Berechnung neuer Lösungen (Individuen) zu steuern (siehe Abschnitt 3.6.4). Die Tabelle 3.1 vermittelt noch einmal einen Überblick der verwendeten Begriffe.

biologischer Begriff	Bedeutung/Entsprechung bei GA
Individuum	eine einzelne Lösung des gegebenen Problems
Population	eine Menge von Lösungen
Eltern	zur Reproduktion ausgewählte Lösungen
Nachkommen / Kinder	aus den Eltern generierte / neugebildete Lösungen
Crossover	Operator, der Eigenschaften der Eltern vermischt
Mutation	Operator, der Eigenschaften eines einzelnen Individuums verändert
Fitness	Lösungsgüte, bezogen auf die Ziele
Generation	Verfahrensiteration
Gen	einzelner Parameter einer Lösung
Allel	Genausprägung (explizite Wertebelegung)

**Tabelle 3.1:** Terminologie der genetischen Algorithmen, nach [Nis97]

### 3.6.3 Der Algorithmus

Bei genetischen Algorithmen handelt es sich um populationsbasierte Algorithmen. Der Algorithmus verwaltet zu jeder Zeit eine Menge an möglichen Lösungen in einer Population. Alle Berechnungen, die der Algorithmus im Laufe einer Optimierung durchführt, basieren auf der jeweils aktuellen Population. Folglich ist es zu Beginn notwendig, eine solche *Startpopulation* zu initialisieren. Üblicherweise ist die Größe der Population in den Ablaufparametern des Algorithmus vorgegeben. Somit müssen entsprechend der Vorgabe eine festgelegte Anzahl an Individuen generiert werden. Die Initialisierung der Startpopulation kann auf zwei Arten erfolgen:

1. zufällige Verteilung der Individuen über den Suchraum
2. manuelles Erzeugen der Individuen

Die zweite Möglichkeit ist vor allem dann sinnvoll einsetzbar, wenn bereits einige Vorkenntnisse oder Vermutungen über die Lage der gesuchten Optima im Suchraum existieren. Dann können die Individuen gezielt in diesen Bereichen des Suchraumes platziert werden. Allerdings birgt dieses Vorgehen eine größere Gefahr vorzeitig in lokalen Optima zu konvergieren. Daher wird bei den meisten Optimierungen, insbesondere bei neuen Problemstellungen, die zufällige Verteilung der Individuen über den Suchraum dem Algorithmus überlassen. Die

bei einigen Optimierungsproblemen zusätzlich gegebenen Nebenbedingungen für die Gültigkeit der Lösungen, z. B. Parameterwertebereiche, müssen selbstverständlich auch bei der Initialisierung der Startpopulation berücksichtigt werden.

Der Algorithmus bewertet im zweiten Schritt die generierten Individuen. Zu diesem Zweck muss eine *Fitnessfunktion* definiert werden, die abhängig von den Optimierungszielen ist. Die Fitnessfunktion definiert daher eine Berechnungsgrundlage für die Bewertung der einzelnen Individuen. Damit entspricht sie aus Sicht klassischer Optimierungsprobleme der Zielfunktion, aus Sicht der Evolution einem Maß für die Überlebensfähigkeit eines Lebewesens. Die Definition der Fitnessfunktion ist eine Abbildung, die jedem Element des Suchraumes eine Maßzahl als Bewertung der Lösungsgüte zuweist:

$$fit : S \rightarrow [0, \infty) \quad (3.2)$$

Nachdem die Fitness jedes Individuums (die Güte jeder Lösung) berechnet ist, können neue Lösungen ermittelt werden. Zu diesem Zweck werden aus der aktuellen Population Lösungen ausgewählt, auf deren Basis dann neue Lösungen berechnet werden. Dieser Vorgang wird als *Selektion* bezeichnet. Der genaue Ablauf ist dabei von dem jeweils ausgewählten Selektionsverfahren abhängig (z.B. Roulette-Selektion, Turnier-Selektion). Einzelne Verfahren werden in Abschnitt 3.6.4.1 näher erläutert. In allen Verfahren erfolgt jedoch die Auswahl der einzelnen Individuen auf Basis ihrer Fitnesswerte. Individuen mit einer höheren Fitness werden mit einer größeren Wahrscheinlichkeit selektiert als solche mit einer geringeren Fitness. Dieses Prinzip entspricht der darwinistischen Vorstellung, dass Individuen mit einer besseren Anpassung an ihren Lebensraum eine höhere Erfolgsaussicht in Bezug auf die Zeugung von Nachkommen haben.

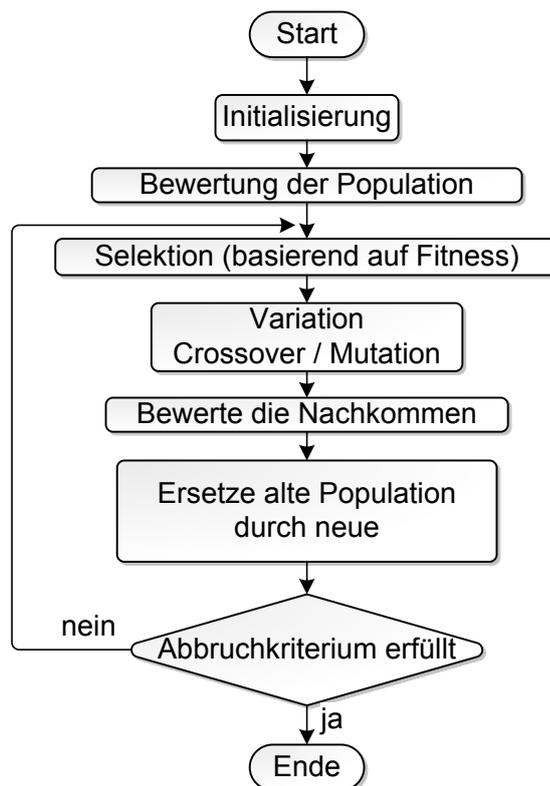
Aus den ausgewählten Individuen, ein Individuum kann dabei auch mehrfach ausgewählt werden (Ziehen mit zurücklegen), werden im Folgenden neue Individuen generiert. Entsprechend dem Vorbild der Genetik werden dabei die Eigenschaften mehrerer Individuen durch geeignete Operatoren „vermischt“. Dieser Vorgang wird als *Reproduktion* bezeichnet. Die Reproduktion kann auf zwei Arten erfolgen. Der größere Teil an Nachkommen wird nach dem Vorbild der Genetik durch Kreuzung mehrerer Individuen erzeugt. Hierbei stehen erneut unterschiedliche Verfahren und Methoden zur Verfügung, mit denen die Eigenschaften der Nachkommen aus denen der Eltern ermittelt werden. Einen Überblick über eine Auswahl an Crossover-Operatoren findet sich in Abschnitt 3.6.4.2. Der zweite, kleinere Teil der Nachkommen wird erzeugt, indem eine oder mehrere Eigenschaften eines einzelnen Individuums mehr oder weniger zufällig verändert werden. Dieser Vorgang wird als *Mutation* bezeichnet. Diese zufälligen Variationen der Eigenschaften einzelner Individuen sind notwendig, um die genetische Vielfalt der Populationen aufrecht zu erhalten, beziehungsweise neue Informationen in den Genpool der Population einzubringen.

Diese so entstandenen neuen Individuen bilden eine neue Population und ersetzen die

bisherige. Ebenso wie in der Startpopulation müssen nun für alle Individuen der neuen Population die Fitnesswerte bestimmt werden. Erreicht der Algorithmus an dieser Stelle ein gegebenes Abbruchkriterium, so terminiert er mit der Ausgabe des größten gefundenen Fitnesswertes und den Eigenschaften des zugehörigen Individuums (den Parametern der zugehörigen Lösung). Andernfalls startet der Kreislauf von Selektion, Reproduktion und Fitnessbewertung erneut.

Ein Abbruchkriterium für genetische Algorithmen ist das Erreichen einer geforderten und bekannten Lösungsgüte. Alternativ werden die Optimierungen in ihrer Laufzeit begrenzt. Dies kann zum einen eine feste Anzahl an Generationen sein, nach denen der Algorithmus in jedem Fall terminiert, oder eine Anzahl an Generationen, in denen der Algorithmus keine bessere Lösung mehr ermitteln konnte, oder aber eine Kombination aus beiden Möglichkeiten.

Der beschriebene Kontrollfluss eines genetischen Algorithmus wird in Abbildung 3.5 noch einmal graphisch dargestellt.



**Abbildung 3.5:** Kontrollfluss eines genetischen Algorithmus nach [BK05]

### 3.6.4 Operatoren

Die Anwendung eines einfachen, genetischen Basisalgorithmus ist relativ simpel und prinzipiell leicht möglich. Jedoch sind die erzielten Ergebnisse häufig nicht von überzeugender Qualität ([GKK04]). Daher ergibt sich zwingend die Notwendigkeit, die verschiedenen Parameter und Operatoren des Algorithmus an das zu betrachtende Problem anzupassen. Der folgende Abschnitt gibt einen Überblick über eine Reihe unterschiedlicher Selektions- und Crossover-Operatoren, die im Rahmen eines genetischen Optimierungsalgorithmus Verwendung finden.

Allerdings lassen sich insbesondere für die Einstellung der Ablaufparameter jedoch keine bindenden Aussagen treffen. Umfangreiche Tests dieser Parameter sind daher nötig (siehe Kapitel 4.3), um den Algorithmus entsprechend an die Problemstellung anzupassen.

#### 3.6.4.1 Selektionsverfahren

In diesem Abschnitt werden unterschiedliche Verfahren vorgestellt, mit denen die Selektion der einzelnen Individuen durchgeführt werden kann. Grundsätzlich sollen bei allen Selektionsverfahren Individuen mit einer guten Fitness gegenüber Individuen mit einer schlechteren Fitness bevorzugt ausgewählt und die durchschnittliche Fitness innerhalb einer Population so kontinuierlich gesteigert werden.

**Stochastisch uniforme Selektion** Ein sehr einfaches Verfahren zur Selektion ist die stochastisch uniforme Selektion [Bak85]. Hierbei werden die einzelnen Individuen auf einer Achse aneinandergereiht. Die Länge des Achsenabschnittes, der dem einzelnen Individuum zugewiesen wird, ist dabei proportional zu dem Anteil der individuellen Fitness an der Gesamtfitness der Population. Anschließend läuft der Algorithmus in immer gleich großen Schritten die Achse entlang und wählt nach jedem Schritt das Individuum, welches dem aktuellen Abschnitt zugeordnet ist. Dabei kann ein Individuum auch mehrfach ausgewählt werden. Um eine gewisse Randomisierung der Selektion zu gewährleisten, wird die Länge des ersten Schrittes variabel gestaltet. Für den ersten Schritt wird die Schrittlänge um einen Zufallswert verkleinert. Abbildung 3.6 zeigt ein Beispiel für die stochastisch uniforme Selektion mit zehn Individuen und einer definierten Schrittweite von 3 für den Auswahlalgorithmus, bei dem eine Auswahl von zehn Individuen getroffen wird. Folgende Individuen werden ausgewählt:  $\{1, 1, 2, 4, 5, 7, 7, 9, 9, 10\}$

**Roulette-Selektion** Bei der Roulette-Selektion handelt es sich um eines der meist benutzten Selektionsverfahren. Die Roulette-Selektion wird auch als fitnessproportionale Selektion bezeichnet [BK05] [Hol92]. Die Auswahl bei der Roulette-Selektion basiert auf der Simulation eines Rouletterades, das in unterschiedlich große Sektoren unterteilt wird. Jedem einzelnen Individuum wird dabei ein Sektor zugeordnet. Die Größe des Sektors ist

dabei vom Fitnesswert des zugehörigen Individuums abhängig. Zur Selektion wird eine Zufallszahl generiert (entspricht der Kugel im Roulette-Spiel), die in einem der Sektoren liegt. Das zu dem Sektor gehörende Individuum wird somit ausgewählt. Gegenüber anderen Selektionsverfahren besitzt die Roulette-Selektion den Vorteil, dass auch Individuen mit einer schlechten Fitness eine Chance haben selektiert zu werden. Die Chance ist zwar vergleichsweise klein, verringert aber für den Algorithmus die Gefahr, vorzeitig in einem lokalen Optimum zu konvergieren. Abbildung 3.7 zeigt ein Beispiel für die Aufteilung der Individuen auf dem Rouletterad.

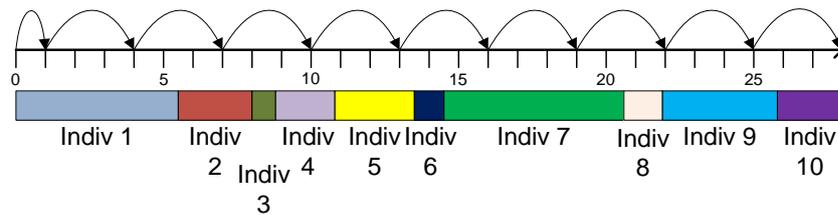


Abbildung 3.6: stochastisch uniforme Selektion

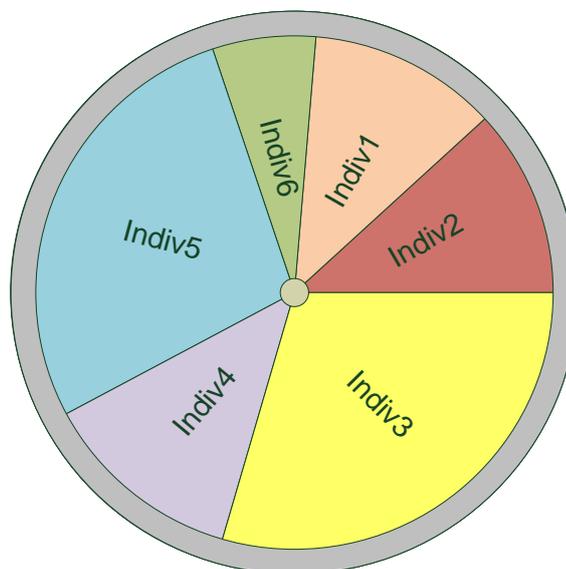


Abbildung 3.7: Roulette-Selektion

**Rest-Selektion** Dieses Selektionsverfahren selektiert die Eltern aufgrund der ganzzahligen Werte ihrer Fitness deterministisch. Anschließend wird eine auf einer Roulette-Selektion basierende Selektion durchgeführt, die auf den Nachkommastellen des Fitnesswertes basiert

[GKK04]. Wenn zum Beispiel der berechnete Fitnesswert eines Individuums 2,3 beträgt, dann wird das Individuum zweimal als möglicher Elter aufgelistet, da der ganzzahlige Anteil 2 beträgt. Nachdem die möglichen Eltern auf diese Weise ermittelt wurden, werden sie stochastisch ausgewählt. Die Wahrscheinlichkeit, dass ein Individuum dabei ausgewählt wird, ist dabei proportional zum Nachkommawert seines Fitnesswertes<sup>1</sup>.

**Turnier-Selektion** Bei der Turnier- oder Wettkampf-Selektion wird eine vorgegebene Anzahl (Turnier-Größe) von Individuen uniform zufällig aus der Population ausgewählt [GKK04]. Aus dieser Auswahl wird dann das Individuum mit der höchsten Fitness selektiert. Der Mindestwert für die Turniergröße *tournamentsize* beträgt also 2, ansonsten entspräche das Selektionsverfahren einer einfachen uniformen Selektion über alle Individuen.

Bei der Turnier-Selektion ist letztlich nur die Reihenfolge der Fitnesswerte entscheidend, der absolute Fitnesswert wird nicht in Betracht gezogen. Die Anzahl der Turniere, die durchgeführt werden, entspricht der Anzahl an Individuen, die selektiert werden sollen. Nachteilig wirkt sich in der Turnier-Selektion allerdings aus, dass die Individuen mit den *tournamentsize* - 1 schlechtesten Fitnesswerten nie selektiert werden [Kin94].

### 3.6.4.2 Crossover-Verfahren

Nach der Selektion der Individuen folgt die Reproduktion, das Erzeugen neuer Individuen. Der folgende Abschnitt stellt einige unterschiedliche Reproduktionsverfahren vor.

**Binärer Crossover** Der *binäre Crossover*-Operator erzeugt als Reproduktionsverfahren zunächst einen binären Zufalls-Bitvektor [BK05]. Die Länge des Vektors entspricht der Zahl der Gene in einem Individuum. Das Kind wird durch eine Kombination der Gene der Eltern gewonnen. Besitzt der Bitvektor den Wert 1, so werden die Gene des ersten Elter ausgewählt, besitzt der Bitvektor den Wert 0, werden die Gene des zweiten Elter ausgewählt (siehe Abbildung 3.8).

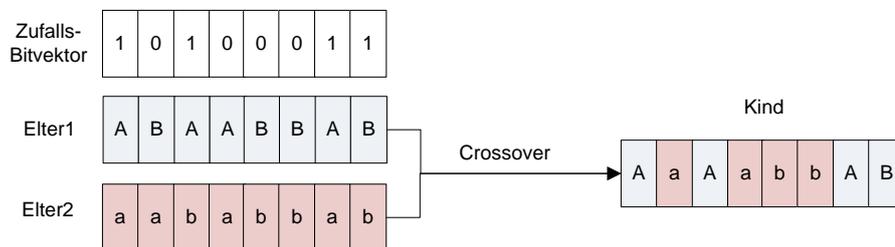


Abbildung 3.8: binärer Crossover

<sup>1</sup>Das hier verwendete Selektionsverfahren beschreibt eine geringfügig abweichende Umsetzung zu dem in [GKK04] beschriebenen Verfahren und entspricht der Implementierung in [mat08].

**1-Punkt Crossover** Einen weiteren Operator zur genetischen Reproduktion stellt der 1-Punkt Crossover dar. Mittels einer ganzzahligen Zufallszahl aus dem Intervall  $[0, x]$ ,  $x =$  Anzahl Gene pro Individuum, wird ein Kreuzungspunkt bestimmt, ab der die Allele der beiden Eltern vertauscht werden [GKK04]. Auf diese Weise können statt einem theoretisch auch zwei Nachkommen gleichzeitig erzeugt werden, indem bis zur Trennstelle Kind 1 (Kind 2) die Allele des ersten (zweiten) Elter erhält und nach der Trennstelle die Werte der Gene von Elter 2 (Elter 1). Abbildung 3.9 zeigt ein Beispiel, bei dem nur ein Kind generiert wird.

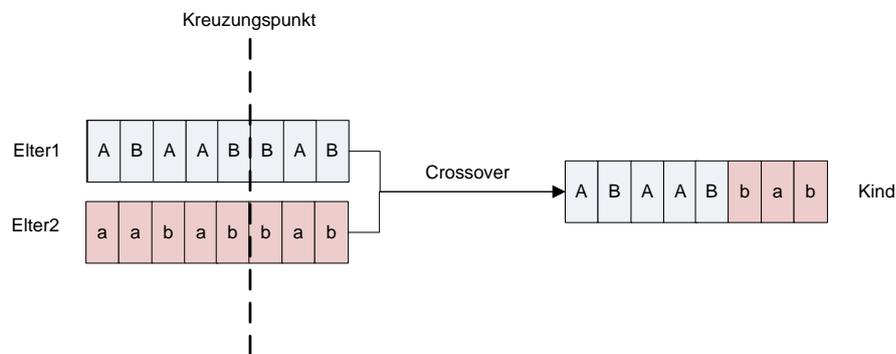


Abbildung 3.9: 1-Punkt Crossover

**2-Punkt Crossover** Beim 2-Punkt Crossover handelt es sich um eine Erweiterung des 1-Punkt Crossover-Operators. Diesmal werden zwei Kreuzungspunkte  $m, n$  bestimmt [GKK04]. Die Generierung eines neuen Individuums erfolgt dann nach folgendem Schema:

- Die Allele an Positionen  $x$ ,  $x < m$  werden vom ersten Elter übernommen.
- Die Einträge an Positionen  $x$ ,  $m < x < n$  werden vom zweiten Elter übernommen.
- Die Einträge an Positionen  $x$ ,  $x > n$  werden vom ersten Eltern übernommen.

Auch hier besteht die Möglichkeit aus dem einen Elternpaar gleich zwei Nachkommen zu erzeugen, indem die Auswahl der Gene der Eltern, die für das neue Individuum übernommen werden, invertiert wird. Abbildung 3.10 zeigt wieder den einfachen Fall mit einem Nachkommen.

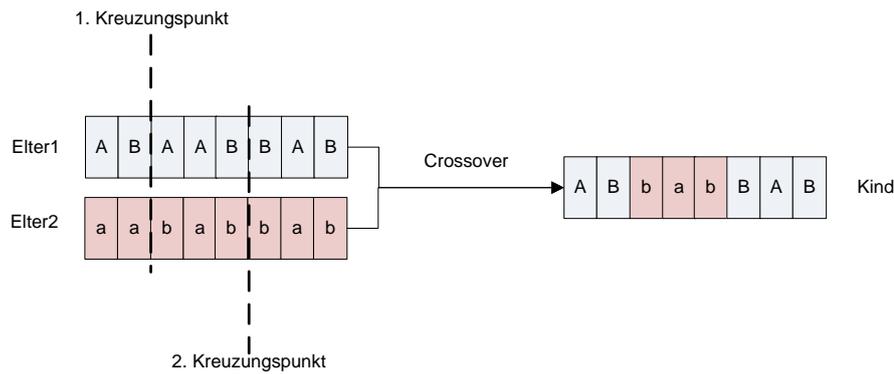


Abbildung 3.10: 2-Punkt Crossover

**Intermediärer Crossover** Durch eine gewichtete Kombination der Gene des Elternpaares erzeugt der intermediäre Crossover-Operator seine Nachfahren [Lip05]. Hierbei handelt es sich um einen Reproduktions-Operator speziell für reelwertige Genausprägungen. Für die Gewichtung der Allele der Eltern kann ein Parameter *Ratio* definiert werden. Die Berechnung der Gene des Kindes erfolgt dann nach der Formel<sup>2</sup>:

$$Kind = Elter1 + rand \cdot Ratio \cdot (Elter2 - Elter1) \quad (3.3)$$

Die Lage der Nachkommen in Relation zu den Eltern wird in erster Linie durch den Wert des Parameters *Ratio* bestimmt. Bei dem Parameter kann es sich entweder um einen Vektor handeln, dessen Länge der Anzahl der Gene eines Individuums entspricht, oder um einen Skalar. Liegen die Werte des Vektors alle innerhalb des Intervalls  $[0, 1]$ , so liegen die Nachkommen innerhalb des Hyperwürfels, der aufgespannt wird, indem die Eltern an gegenüberliegenden Ecken platziert werden. Liegt einer oder mehrere Werte des Vektors *Ratio* außerhalb des Intervalls, so befindet sich ein erzeugtes Kind möglicherweise außerhalb des aufgespannten Würfels. Handelt es sich bei dem Parameter *Ratio* um einen Skalar, so befinden sich die erzeugten Kinder auf der Geraden, die beide Eltern miteinander verbindet.

Die Abbildung 3.11 verdeutlicht den Zusammenhang zwischen den Eltern und den generierten Kindern anhand eines Beispiels für einen drei-dimensionalen Suchraum.

Die Eltern haben dabei die folgenden Gen-Ausprägungen:

- Elter 1 (blau): 1/1/1
- Elter 2 (rot): 2/2/2

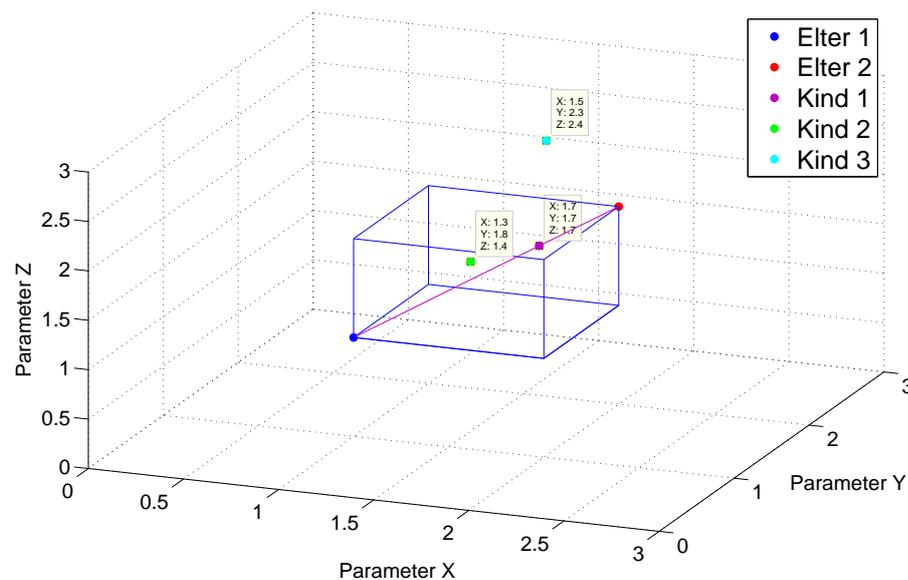
<sup>2</sup>Die hier verwendete Berechnungsgrundlage erweitert die ursprüngliche Formel aus [Lip05] um den *Ratio*-Parameter und entspricht der Implementierung in [mat08]

Der Ratio-Parameter für den Crossover-Operator sei im Falle von Kind 1 (lila) der Skalar  $n = 0,7$ , im Falle von Kind 2 (grün) der Vektor  $m = \begin{pmatrix} 0,3 \\ 0,8 \\ 0,4 \end{pmatrix}$  und für Kind 3 (türkis) der

Vektor  $o = \begin{pmatrix} 1,5 \\ 1,3 \\ 1,4 \end{pmatrix}$ . Somit ergeben sich folgende Allele für die Kinder:

- Kind 1: 1,7/1,7/1,7
- Kind 2: 1,3/1,8/1,4
- Kind 3: 2,5/2,3/2,4

Kind 1 liegt auf der direkten Verbindung zwischen den beiden Eltern. Kind 2 liegt innerhalb des durch die Eltern aufgespannten Würfels, Kind 3 liegt außerhalb des Würfels. Die Randomisierung durch den Parameter  $rand$  wurde in dem Beispiel nicht berücksichtigt.



**Abbildung 3.11:** Beispiel eines intermediären Crossover mit drei Parametern

**Heuristischer Crossover** Ähnlich wie beim intermediären Crossover werden beim heuristischen Crossover die Kinder durch eine gewichtete Kombination der Eltern ermittelt [GKK04]. Auch der heuristische Crossover ist ein Operator speziell für Gene, die reelwertige Allele besitzen. Das berechnete Kind liegt auf der Geraden, die die beiden Eltern

verbindet. Es liegt in der Nähe des Elters mit der höheren Fitness auf der dem anderen Elter abgewandten Seite. Wie weit das Kind vom besseren Elter entfernt liegt, kann mittels eines Parameters *Ratio* bestimmt werden. Die Berechnung erfolgt nach der Formel<sup>3</sup>:

$$Kind = Elter2 + Ratio \cdot (Elter1 - Elter2) \quad (3.4)$$

Elter2 bezeichnet in diesem Fall den Elternteil mit der niedrigeren Fitness. Der Standardwert des Ratio-Parameters beträgt 1,2.

**Arithmetischer Crossover** Hier erfolgt die Berechnung der Kinder auf Basis eines gewichteten Mittels der Eltern [GKK04]. Die Gewichtung zwischen den Eltern erfolgt dabei durch das Ziehen eines Zufallswertes. Die Berechnung der Allele für einen Nachkommen wird anhand folgender Formel durchgeführt:

$$Kind = Rand \cdot Elter1 + (1 - Rand) \cdot Elter2; \quad (3.5)$$

Rand bezeichnet einen Zufallswert aus dem offenen Intervall ]0,1[. Auch der arithmetische Crossover-Operator ist, wie die beiden vorherig beschriebenen Operatoren, speziell für reelwertige Gene konzipiert.

### 3.6.5 Weitere Anpassungsmöglichkeiten

Neben der Auswahl der beschriebenen Operatoren zur Steuerung der genetischen Reproduktion lassen sich eine Reihe zusätzlicher Erweiterungen einsetzen um genetische Algorithmen an die unterschiedlichsten Problemstellungen anzupassen. Weiterhin kann durch diverse Ablaufparameter Einfluss auf den Optimierungsverlauf genommen werden. Im Folgenden werden einige dieser Parameter und Erweiterungen vorgestellt.

**Populationsgröße** Die Populationsgröße stellt einen zentralen Parameter mit großem Einfluss auf die Laufzeit und die Ergebnisse einer Optimierung durch genetische Algorithmen dar [GKK04]. Wird die Populationsgröße zu niedrig gewählt, schlägt sich dies in einer vorzeitigen Konvergenz und somit einer suboptimalen Lösung des Optimierungsproblems nieder. Wird die Populationsgröße zu hoch angesetzt, ist dies zwar positiv für die genetische Vielfalt innerhalb der Population, insbesondere aber bei Problemstellungen, bei denen die Auswertung der einzelnen Fitnesswerte sehr zeitaufwendig ist (wie im vorliegenden Fall der Simulation eines Bussystems), führt eine zu große Population zu unnötig hohen Laufzeiten.

**Elite-Übernahme** Wird eine Population vollständig durch die Population der Nachfolge-Generation ersetzt, kann dies dazu führen, dass sowohl die Bewertung des besten Individuums als auch die durchschnittliche Bewertung der neuen Population schlechter ist als die

<sup>3</sup>Die hier verwendete Berechnungsformel ersetzt in der ursprünglichen Formel aus [GKK04] den Zufallswert durch den *Ratio*-Parameter und entspricht der Implementierung in [mat08]

der Elterngeneration. Das Konzept der Elite-Übernahme sieht vor, eine festgelegte Anzahl der besten Individuen direkt in die nächste Population zu übernehmen. Damit soll sichergestellt werden, dass bereits gefundene gute Lösungskandidaten durch die randomisierten Selektions-, Crossover- und Mutationverfahren überleben und nicht wieder verworfen werden. Ein zu hoher Wert des Parameters, also eine zu umfangreiche Übernahme von (guten) Individuen, führt in der Regel zu einer vorzeitigen, suboptimalen Konvergenz des Algorithmus [SHF94]. Daher werden üblicher Weise nur wenige Individuen direkt in die nächste Population übernommen.

**Verhältnis zwischen Crossover und Mutation** Um eine geeignete Steuerung der Mutationen zu ermöglichen, ist es notwendig, den Anteil der Individuen festzulegen, die durch genetische Kreuzung entstehen [GKK04] [Kin94] [Nis97]. Damit wird verhindert, dass der Mutationsanteil der Population unbrauchbar große Werte annimmt. Üblicherweise wird der Anteil der Population, der ausschließlich durch Anwendung von Crossover-Operatoren erzeugt werden soll, durch Angabe eines Parameters (*Crossover Fraction*) spezifiziert. Als Rest der Population ergibt sich implizit der Anteil Individuen der durch Mutation erzeugt werden soll.

**Fitnesszuweisung** In der bisherigen Darstellung entsprach der Fitnesswert eines Individuums dem aus der Bewertung der Lösungen durch die Zielfunktion gewonnen Wert. Genau genommen muss jedoch zwischen dem Wert der Zielfunktion, die vielfach ebenfalls als Fitness bezeichnet wird, und der internen Fitnesszuweisung des Algorithmus unterschieden werden. Letztere bezeichnet die Fortpflanzungswahrscheinlichkeit eines Individuums und wird aus dem Zielfunktionswert des einzelnen Individuums und den Zielfunktionswerten aller anderen Individuen des entsprechenden Selektionspools durch eine Skalierungsfunktion bestimmt [Poh98]. Generell kann zwischen drei Ansätzen unterschieden werden:

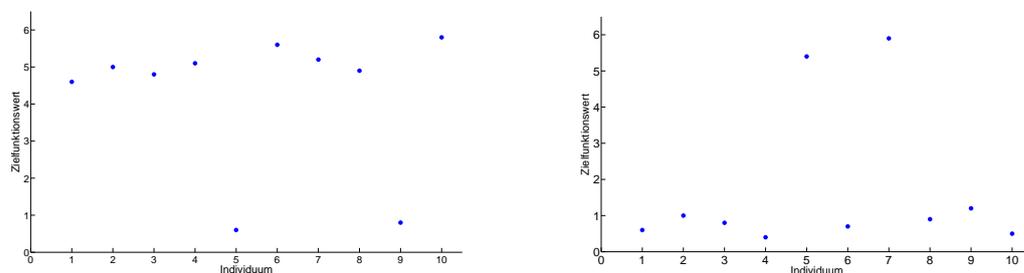
- proportionale Fitness-Skalierung
- reihenfolgebasierte Fitness-Skalierung (rank-based-fitness-scaling)
- multikriterielle Fitnesszuweisung

Bei der proportionalen Zuweisung wird jedem Individuum ein Wert zugewiesen, der proportional zu seinem Zielfunktionswert ist. Hierbei kann es jedoch, wie im Folgenden dargestellt, zu zwei grundlegenden Problemen kommen (siehe Abbildung 3.12):

Das linke Diagramm der Abbildung 3.12 zeigt Individuen, deren Zielfunktionswerte, bis auf zwei Ausnahmen, im Sinne einer Maximierung ähnlich gut sind. Würden die Selektionswahrscheinlichkeiten proportional zum Zielfunktionswert berechnet, würden sich die Wahrscheinlichkeiten ebenfalls kaum unterscheiden. Die Auswahl unter den vielen guten Individuen würde somit einer zufälligen Auswahl recht nahe kommen. Diese Situation wird

als Stagnation bezeichnet und durch den sehr geringen Selektionsdruck<sup>4</sup> verursacht, der durch die proportionale Skalierung erzeugt wird.

Im rechten Diagramm der Abbildung 3.12 haben zwei der dargestellten Individuen einen sehr viel besseren Zielfunktionswert als die übrigen, die alle in etwa gleich auf liegen. Eine proportionale Skalierung der Fitnesszuweisungen würde dazu führen, dass die beiden guten Individuen eine sehr viel höhere Fortpflanzungswahrscheinlichkeit haben als die übrigen Individuen. Somit würden die beiden guten Individuen sehr viele Nachkommen erzeugen, die schlechten Individuen hätten hingegen kaum eine Chance zur Reproduktion. Als Resultat dieses Sachverhaltes würde die genetische Vielfalt in der Population sehr rasch verloren gehen. Am Ende stünde eine vorzeitige Konvergenz des Algorithmus. Die Ursache läge in diesem Fall am zu hohen Selektionsdruck, der durch die proportionale Fitnesszuweisung erzeugt würde. Ideal ist demnach eine Fitnesszuweisung, die einerseits nicht zu einer



**Abbildung 3.12:** Probleme bei proportionaler Fitness-Skalierung

übermäßigen Nachkommenschaft weniger guter Individuen führt, andererseits aber eine ausgeprägte Unterscheidung der Fitnesswerte vieler guter Individuen ermöglicht.

Eine solche Zuweisung ermöglicht die reihenfolge- oder rangbasierte Skalierung der Fitnesswerte. Hierbei wird der Selektionspool nach den Zielfunktionswerten sortiert. Die Fitness, die einem Individuum zugewiesen wird, berechnet sich ausschließlich durch die Position des einzelnen Individuums in dieser sortierten Liste. Durch die rangbasierte Zuweisung wird eine gleichmäßige Skalierung der Fitness über den Selektionspool erreicht. Die Probleme, wie sie bei einer proportionalen Skalierung auftreten können, werden vermieden.

## 3.7 Partikel Schwarm Optimierung

### 3.7.1 Grundlagen

In Schwärmen lebende Tiere (wie Vögel oder Fische) sind in der Lage, durch Kooperation und Informationsaustausch ihre Effizienz bei der Futtersuche signifikant zu steigern. Zu beobachten ist, dass sich die Individuen zunächst über einen gewissen Suchraum verteilen.

<sup>4</sup>Selektionsdruck: Wahrscheinlichkeit der Auswahl des besten Individuums im Vergleich zur durchschnittlichen Selektionswahrscheinlichkeit aller Individuen des Selektionspools

Findet eines der Tiere dann eine Futterstelle, so bewegt sich der gesamte Schwarm auf diese Futterstelle zu und profitiert somit vom Fund des einzelnen Tiers. Dieses Verhalten inspirierte Kennedy und Eberhart zu einem weiteren Optimierungsverfahren, der Partikel Schwarm Optimierung (engl.: Particle Swarm Optimization, PSO)[KE95].

### 3.7.2 Der Algorithmus

Den Kern des Optimierungsverfahrens bilden ähnlich wie beim ACO-Verfahren eine Menge gleichberechtigter und kommunizierender Agenten. Diese Agenten, die in ihrem Verhalten Vögeln bzw. Fischen nachempfunden worden sind, werden als Partikel bezeichnet.

Das Verfahren basiert auf der Idee, dass die einzelnen Partikel, die auf der Suche nach dem globalen Optimum sind, durch den Suchraum fliegen und über den Verlauf bzw. Erfolg ihrer Suche untereinander Informationen austauschen.

Wie der Ansatz zu genetischen Algorithmen verwaltet auch der PSO-Algorithmus eine Population für seine Berechnungen. Jeder einzelne in der Population enthaltene Partikel repräsentiert dabei einen Punkt im Suchraum und somit eine mögliche Lösung. Bei der Initialisierung des Algorithmus wird eine festgelegte Anzahl an Partikeln zufällig über den gegebenen multi-dimensionalen Suchraum verteilt. Zusätzlich werden sogenannte Bewegungsvektoren für jeden Partikel ebenfalls randomisiert erzeugt. Diese spezifizieren Richtung und Geschwindigkeit, mit der sich der Partikel aktuell im Suchraum bewegt. Anschließend startet die Suche nach dem Maximum einer Bewertungsfunktion, die identisch der Fitnessfunktion bei genetischen Algorithmen, jedem Punkt des Suchraumes eine Bewertungszahl zuweist. Die Suche wird realisiert, indem sich die Partikel durch den Suchraum bewegen. Die vorliegende Analogie zwischen dem Algorithmus, der eine maximale Bewertung einer Lösung sucht, und den Vogelschwärmen, die auf der Suche nach dem Ort mit dem meisten Nahrungsangebot sind, ist offensichtlich. Die Bewegungen eines Partikels hängen dabei im Prinzip von drei Faktoren ab. Erstens von der aktuellen Geschwindigkeit und Richtung, mit der sich der Partikel bewegt, zweitens von der Lage guter, bereits gefundener Lösungen, die der Partikel selbst gefunden hat, und schließlich von der besten Lösung, die der Gesamtschwarm gefunden hat. Auch hier liegt die Analogie zum natürlichen Verhalten von Vogel- oder Fischeschwärmen auf der Hand. Kennzeichnend für das Verhalten von Schwärmen ist, dass sowohl die einzelnen Individuen eines Schwarmes für sich individuell Entscheidungen treffen können, diese Entscheidungen aber vom Verhalten des übrigen Schwarms beeinflusst werden können. Im Algorithmus werden diese beiden Aspekte realisiert, indem die Berechnung der Bewegungen des einzelnen Partikels sowohl vom bisherigen persönlichen Optimum als auch vom aktuellen globalen Optimum (personal best, global best) abhängig sind.

Zu diesem Zweck speichert jeder Partikel die beste Lösung, die er selbst bisher gefunden hat (inkl. deren Bewertungsmaßzahl). Zusätzlich wird global die beste bisher von allen

Partikeln gefundene Lösung gespeichert. Die Aktualisierung der Partikelbewegung und Auswertung der Zielfunktion erfolgen rundenbasiert, d.h. in jeder Iteration werden die Lage und die Bewegungen für jeden Partikel neu berechnet. Dadurch bewegt sich der einzelne Partikel auf sein persönliches, aktuelles Optimum und das aktuelle globale Optimum zu. Vorteilhaft bei der PSO-Optimierung ist, dass die Partikel in gewissen Grenzen weitgehend autonom agieren. Es muss also keine aufwändige, zentrale Steuerung im Algorithmus implementiert werden. Eine graphische Darstellung des beschriebenen Kontrollflusses der Partikel Schwarm Optimierung zeigt die Abbildung 3.13

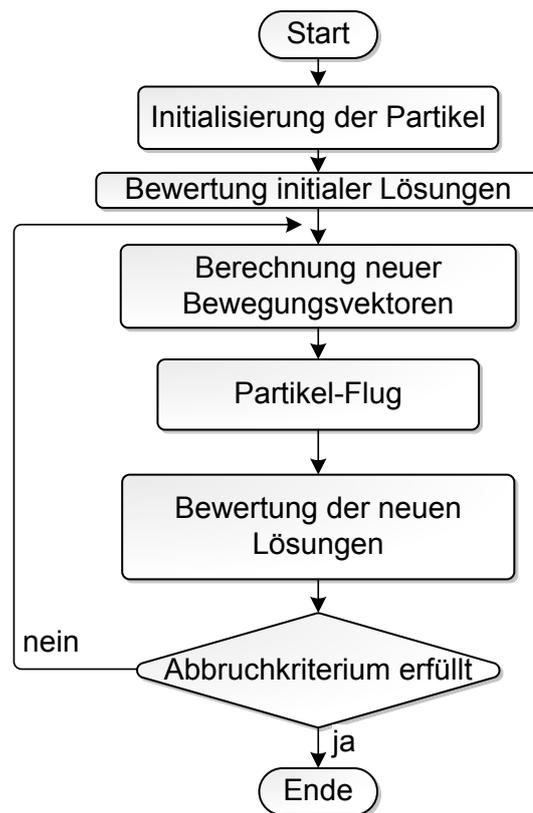


Abbildung 3.13: Schematischer Ablauf PSO nach [BK05]

### 3.7.3 Berechnung der Partikelbewegungen

Auf ihrer Suche nach dem globalen Optimum bewegen sich die Partikel des Schwarms durch den Suchraum. Die Aktualisierung der jeweiligen Position im Raum und die Berechnung der neuen Geschwindigkeit und Richtung eines einzelnen Partikels erfolgt rundenbasiert. Formal errechnet sich die Aktualisierung des Bewegungsvektors  $vel$  für die Bewegung des Partikels von Iteration  $T$  zu Iteration  $T + 1$  anhand von Gleichung 3.6 [SE98b].

$$vel = iwt(i) * vel + ac1 * r1 * (pbest - pos) + ac2 * r2 * (gbest - pos) \quad (3.6)$$

Die einzelnen Bestandteile der Formel definieren sich wie folgt:

- *pbest* bezeichnet die beste bisher vom Partikel selbst gefundene Lösung; *gbest* bezeichnet die beste unter allen Lösungen, die bisher von Gesamtschwarm gefunden worden sind.
- $iwt(i)$  ist ein Trägheitsgewicht (engl.: inertia weight), mit dem der Einfluss gesteuert wird, den die bisherige Bewegung des Partikels auf ihre Neuberechnung ausübt. Je höher der Wert von  $iwt(i)$  ist, desto stärker tendiert der Partikel dazu, neue Bereiche des Suchraums zu explorieren. Verbreitet sind vor allem zwei Realisierungen der Trägheitsgewichtung [SE98a]. Die eine Möglichkeit besteht darin, den Einfluss der aktuellen Bewegung als Konstante in die Berechnung einfließen zu lassen. Die zweite Möglichkeit sieht ein schrittweises Dekrementieren dieses Einflusses im Laufe der Optimierung vor, ähnlich der sinkenden Temperatur des Simulated Annealing. Letztere ist die häufiger verwendete und im Allgemeinen die effektivere Variante [SE98b].
- *ac1* und *ac2* werden als Beschleunigungskoeffizienten bzw. kognitiver und sozialer Parameter bezeichnet. Sie spezifizieren den Einfluss der individuell beziehungsweise global gefundenen, besten, bisherigen Lösung bei der Berechnung der neuen Bewegung.
- *r1* und *r2* sind zwei gleichverteilte Zufallsvariablen aus den Intervall[0, 1].

Die neue Position *pos* eines einzelnen Partikels errechnet sich dann einfach aus der Addition (Gleichung 3.7) der bisherigen Position und des neu berechneten Bewegungsvektors aus Gleichung 3.6 [SE98b].

$$pos = pos + vel \quad (3.7)$$

Es ist offensichtlich, dass der Aufwand für eine Implementierung eines PSO-Verfahren im Vergleich zu genetischen Algorithmen mit ihren zahlreichen Auswahlverfahren, Wahrscheinlichkeitsberechnungen und Kreuzungsoperatoren deutlich geringer ist.

### 3.7.4 Trägheitsgewicht

Die Realisierung des Trägheitsgewichtes erfolgt vielfach so, dass die Gewichtung der aktuellen Bewegung im Laufe der Optimierung fortlaufend bis zu einem gewissen Wert vermindert wird [ZYH03]. Dieser Abschnitt erläutert diese Standard-Variante zur Realisierung des Trägheitsverhalten der Partikel.

Die Berechnung des aktuellen Trägheitswertes in Iteration *i* erfolgt nach Formel 3.8.

$$iwt(i) = \frac{iw2 - iw1}{iwe - 1} * (i - 1) + iw1 \quad (3.8)$$

Dabei bezeichnet *iw1* den Trägheitswert zu Beginn der Optimierung, *iw2* spezifiziert die untere Grenze, bis zu der der Parameter absinken können soll. Ist die Optimierung bereits

so weit fortgeschritten, dass sich der Algorithmus in einer Iteration  $i > iwe$  befindet, wobei  $iwe$  die Iteration bezeichnet, in der der finale Wert für die Trägheitsgewichtung erreicht wurde, so wird dieser finale Wert für alle folgenden Iterationen übernommen und nicht mehr verändert. Es ergibt sich für alle Iterationen  $i > iwe$

$$iwt(i) = iw2 \quad (3.9)$$

### 3.7.5 Anpassungen

Im Vergleich zu genetischen Algorithmen ist die notwendige Anpassung der Ablaufparameter bei PSO-Optimierungen deutlich weniger umfangreich. Im Folgenden werden die wichtigsten Einflussparameter des Verfahrens vorgestellt.

**Größe des Schwarms** Ähnlich wie bei genetischen Algorithmen ist die Populations- bzw. Schwarmgröße mitentscheidend für die Effektivität des Algorithmus. Generell kann davon ausgegangen werden, dass bei einem größeren Schwarm die Abdeckung des Suchraumes besser gewährleistet ist als bei einem kleinen Schwarm. Es ist zu erwarten, dass ein großer Schwarm weniger Iterationen benötigt, um ein Optimum zu finden, als ein vergleichsweise kleinerer. Allerdings ist insbesondere bei Problemstellungen, bei denen die Auswertung der aktuellen Partikelpositionen sehr zeitaufwändig ist, von einer zu hohen Anzahl von Partikeln abzusehen, da dies sonst immensen Einfluss auf die Laufzeit der Optimierung hat und diese unnötig verlängert.

**Trägheitsgewichtung** Die bereits oben vorgestellte Berechnung des Trägheitsgewichts ist neben der Populationsgröße die wichtigste Einflussgröße zur Steuerung der algorithmischen Abläufe der Partikel Schwarm Optimierung. Bei der Realisierung in Form eines absinkenden Parameterwertes müssen insgesamt drei Werte sinnvoll belegt werden:

- der initiale Startwert zu Beginn der Optimierung
- der Endwert, bis zu dem ein Absinken des Parameterwertes umgesetzt werden soll
- die Nummer der Iterationsstufe, in der dieser Wert erreicht werden soll

Ein zu hohes oder ein zu langsam sinkendes Trägheitsmaß führt dazu, dass die Partikel sehr stark von ihrer ursprünglichen Bewegung beeinflusst werden. Das bedeutet, dass der Einfluss des Schwarms auf den einzelnen Partikel relativ gering ist und die Konvergenz des Schwarms in Richtung des globalen Optimums sehr langsam verlaufen würde. Somit wird in der Umgebung der aktuell besten Lösung nur sehr spärlich nach weiteren Verbesserungen gesucht. Ein zu geringes Trägheitsgewicht veranlasst die Partikel hingegen dazu, ihre Bewegungsrichtung recht schnell zu ändern. In der Folge lassen sie sich vom restlichen Schwarm relativ stark beeinflussen und bewegen sich schnell in Richtung der aktuell besten

---

Lösung. Dies kann aber dazu führen, dass ein globales Optimum gar nicht gefunden wird, weil sich die Suche zu schnell auf einen zu kleinen Bereich des Suchraums konzentriert (vorzeitige Konvergenz).

**Beschleunigungskoeffizienten** Werden die Beschleunigungskoeffizienten für die Berechnung der neuen Partikel Bewegungen variiert, ergibt sich eine weitere Möglichkeit Einfluss auf den Optimierungsverlauf zu nehmen. Durch die Koeffizienten kann gesteuert werden, in welchem Umfang die jeweils persönlich gefundene beste Lösung eines Partikel und die beste global im Schwarm gefundene Lösung die neuen Bewegungsrichtungen der Partikel beeinflussen (siehe Formel 3.8).

**Abbruchkriterien** Die Abbruchkriterien, die für das PSO-Verfahren eingesetzt werden, sind wieder vergleichbar mit denen eines genetischen Algorithmus. Entweder terminiert der Algorithmus bei Erreichen einer vorgegeben Lösungsgüte oder nach einer festgelegten Anzahl an Iterationen oder wenn er über eine bestimmte Anzahl an Iterationen keine bessere Lösung mehr gefunden hat.

**Umgang mit Wertebereichsgrenzen** Bei Problemstrukturen, deren Parameter über einen begrenzten Wertebereich verfügen, lassen sich unterschiedliche Ansätze realisieren. Insbesondere sind zwei Ansätze denkbar, wenn ein Partikel bei seinem Flug durch den Suchraum eine solche Grenze erreicht. Ein Überschreiten der Grenze ist nicht sinnvoll, da dies zu ungültigen Lösungen des Problems führen würde. Stattdessen besteht die Möglichkeit den Partikel bei Erreichen der Wertebereichsgrenze abprallen zu lassen und in den Suchraum zu reflektieren. Andererseits kann der Parameter, dessen Wertebereichsgrenze erreicht wurde, für diese Iterationsrunde auf den definierten Grenzwert fixiert werden.

### 3.8 Bewertung und Auswahl der Verfahren

Die einfachen, algorithmischen Ansätze des Hill-Climbing-Algorithmus sowie der Tabu-Suche werden in dieser Arbeit für die Suche nach einer Busparameterkonfiguration, die zu einer größtmöglichen Beeinträchtigung der Signalintegrität des zu analysierenden Kfz-Bussystems führt, nicht weiter betrachtet. Für beide Ansätze ist nicht zu erwarten, dass sie bei einer vergleichbaren Zahl an Simulationen ähnlich gute Ergebnisse erreichen wie der Simulated Annealing Algorithmus [Gü07]. Simulated Annealing basiert zwar auf dem gleichen Grundgedanken, die Umsetzung ist aber deutlich robuster gestaltet. Die Möglichkeit während der Suche auch Verschlechterungen in Kauf zu nehmen und den Suchbereich langsam zu verkleinern, ermöglichen einen Einsatz des Optimierungsalgorithmus auch bei komplexeren Problemstellungen. Da Simulated Annealing bereits in [Gü07] zur Anwendung kam, dort eine Analyse und insbesondere der Vergleich zu anderen Optimierungsal-

gorithmen jedoch fehlt, wird der Algorithmus als Vergleichsbasis für die neuen Verfahren eingesetzt und in die Analysen einbezogen.

Das Anforderungsprofil der Parametervariation innerhalb der Bussystemanalyse entspricht nicht den Voraussetzungen für eine erfolgreiche Realisierung des ACO-Algorithmus (siehe Abschnitt 3.5.1). Bei dem vorliegenden Problem handelt es sich weder um ein kombinatorisches Optimierungsproblem noch existiert eine konstruktive Methode, mit der sich die Lösungskandidaten erzeugen lassen. Aus diesen Gründen ist der sonst vielfach sehr erfolgreich eingesetzte Optimierungsansatz der Ameisenkolonieoptimierung hier nicht einsetzbar. Sowohl genetische Algorithmen als auch die Partikel Schwarm Optimierung stellen jeweils ein weiteres komplexes und leistungsfähiges Optimierungsverfahren dar. Aufgrund ihrer Populationsbasiertheit besitzen beide Verfahren die Fähigkeit, einen vorgegebenen multidimensionalen Suchraum ausgehend von mehreren Positionen parallel zu durchsuchen. Insbesondere bei sehr komplexen Optimierungsaufgaben sind hier Vorteile gegenüber Algorithmen zu erwarten, die ihre Suche nur auf einer Lösungsinstanz aufbauen. Zusätzlich sollte es aufgrund ihrer Struktur für beide Algorithmen mit vergleichsweise geringem Aufwand möglich sein, die Berechnungen während der Optimierungen zu parallelisieren, um so vor allem die Laufzeitperformance der Optimierungen deutlich zu steigern. Dies ist im Hinblick auf den Einsatz innerhalb der Bussystemanalyse ein nicht unwesentlicher Aspekt, da insbesondere die Bewertung der unterschiedlichen Parameterkonfiguration sehr zeitaufwändig ist. Aus den angeführten Gründen scheinen beide Algorithmen für einen Einsatz zur Parametervariation innerhalb der Bussystemanalyse gut geeignet zu sein.

Im nächsten Kapitel werden die implementierten Verfahren ausgiebigen Analysen unterzogen, um zum einen eine möglichst gute Anpassung der Algorithmen an die Aufgabe zu gewährleisten und zum anderen, um die jeweilige der Algorithmen näher zu untersuchen.

# 4 Anpassen der Algorithmen

In den folgenden Abschnitten wird beschrieben, wie die in Kapitel 3 vorgestellten und ausgewählten Algorithmen bestmöglich an die Aufgabenstellung der Parametervariation innerhalb der automatisierten Analyse von Kfz-Bussystemen (siehe Abschnitt 2.2) angepasst werden. Zu diesem Zweck werden die Algorithmen im Wesentlichen an zwei Bus-Modellen unterschiedlicher Komplexität getestet. Untersucht wird der Einfluss einzelner Ablaufparameter und mögliche Performanceverbesserungen durch gezielte Erweiterungen der jeweiligen Algorithmen. Die resultierenden Ergebnisse werden im Anschluss auf die erreichte Ergebnisgüte und die benötigte Zahl an Simulationen während der Analyse verglichen und bewertet.

## 4.1 Testmodelle

Zunächst erfolgt die Definition der Testmodelle, die für die Analyse der Algorithmen benötigt werden. Im Anschluß beginnt die eigentliche Anpassung der drei ausgewählten Algorithmen. Dafür werden zwei unterschiedlich komplexe Bus-Modelle erstellt. Die Unterschiede in ihrer Komplexität soll sicherstellen, dass die erfolgte Anpassung der Algorithmen möglichst unabhängig von der Modellkomplexität ist und nicht nur für Modelle mit gleichem Komplexitätsgrad zutrifft.

An dieser Stelle sei darauf hingewiesen, dass die beiden Modelle ausschließlich dazu dienen, die Algorithmen an die Problemstruktur anzupassen. Es wird nicht der Anspruch erhoben, aussagekräftige Ergebnisse für die Simulation von Kfz-Bussystemen zu erhalten oder kritische Bus-Parameter zu ermitteln.

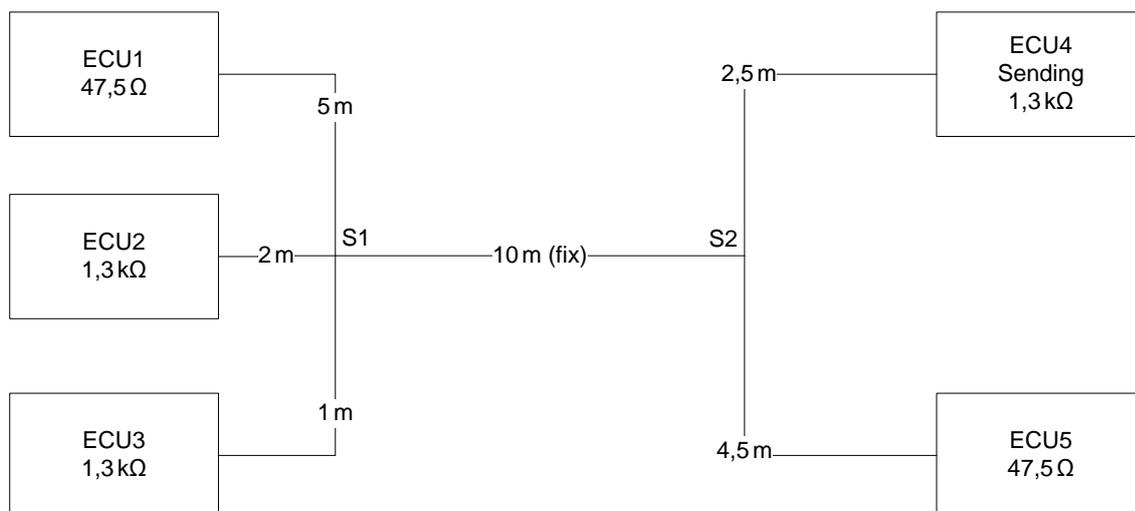
### 4.1.1 Modell 1

Das erste Modell stellt ein von der Topologie her relativ kleines Bussystem dar (siehe Abbildung 4.1). Es verfügt über fünf Teilnehmer (ECUs), die über zwei passive Sterne (engl.: passive Star) miteinander verbunden sind. Für die Analysen der Algorithmen werden folgende Variationsparameter des Bus-Modells definiert:

1. Die Länge der einzelnen Leitungen von den Splices zu den ECUs. Der gültige Wertebereich beträgt jeweils  $\pm 10\%$  vom in der Abbildung 4.1 angegebenen Soll-Wert. Die Länge der Verbindungsleitung zwischen den beiden Splices ist fest und wird nicht variiert.
2. Die Größe des Terminierungswiderstandes an jedem einzelnen Teilnehmer wird ebenfalls variiert. Die zulässigen Werte liegen innerhalb eines  $\pm 10\%$ -Intervalls ausgehend vom angegebenen Vorgabewert.
3. Der Abstand der Leitung zur Massefläche wird jeweils pro Leitung variiert. Auch hier stammen die gültigen Wertebelegungen aus einem  $\pm 10\%$  Intervall um den Vorgabewert  $0,005\text{ m}$ . Bei dieser Variation wird diesmal auch die Verbindungsleitung zwischen den beiden Splices mit einbezogen.

Insgesamt ergibt sich eine Anzahl von 16 Variationsparametern, deren Werte durch die Algorithmen während der Bus-Analyse variiert und so bestimmt werden müssen, dass die zugehörige MSE-Bewertung der Wertekombination maximal wird.

Als sendender Transceiver wird innerhalb des Testmodells ECU 4 ausgewählt, da es sich dabei um eine der Komponenten handelt, die sich im „Innern“ des Busaufbaus befindet und nicht an einem Ende der Topologie. Die sich aus den initialen Wertbelegungen der variablen Parameter (siehe Abbildung 4.1) ergebenden Fehlerwerte werden in Tabelle 4.1 dargestellt. Diese Fehlerwerte stellen den Ausgangspunkt der Optimierungen der Busparameter dar. Es werden die Abweichungen zwischen den empfangenen Signalen an jeder ECU mit dem angepassten Referenzsignal der Daten, die von ECU 4 gesendet werden, verglichen (siehe Abschnitt 2.2.4). Für die Signale, die wieder beim Sender ECU 4 ankommen, ergeben sich mit  $MSE = 0,527405$  die größten Abweichungen.



**Abbildung 4.1:** Topologie von Modell 1

Bus ECU	MSE
1	0,048621
2	0,174994
3	0,087108
4	0,527405
5	0,049125

**Tabelle 4.1:** Modell1 - Fehlerwerte mit initialen Parameterwerten

### 4.1.2 Modell 2

Ein deutlich komplexeres Bussytem stellt das Modell2 dar (siehe Abbildung 4.2). Die Anzahl der Teilnehmer am Bus wird gegenüber Modell 1 auf elf erhöht. Verbunden werden die einzelnen ECUs über nun insgesamt fünf Knotenpunkte. Außer der Zahl der angeschlossenen Komponenten ist auch die Gesamtlänge des Bussystems deutlich größer.

Die Werte der folgenden Parameter sind in der Modell-Vorlage variabel ausgelegt:

1. Die Länge der einzelnen Verbindungsleitungen wird um  $\pm 10\%$  vom jeweiligen Sollwert, der in Abbildung 4.2 spezifiziert ist, variiert. Im Gegensatz zu Modell1 wird nun auch die Länge der Verbindungen zwischen den Splices mit in die Variation einbezogen.
2. Der Wert der Widerstandsterminierung an jedem einzelnen Bus-Teilnehmer wird ebenfalls - wie in Modell 1 - wieder in einem  $\pm 10\%$ -Intervall um den Vorgabewert variiert.
3. Schlussendlich wird auch der Abstand der einzelnen Leitung zur Massefläche mit  $\pm 10\%$  um den Vorgabewert (0,005 m) variiert.

Somit ergibt sich insgesamt eine Anzahl von 41 Variationsparametern für die Optimierungen des zweiten Modells.

Die Funktion des Senders innerhalb von Modell2 übernimmt wieder ECU 4. Die Fehlerwerte, die sich aus den initialen Parameterwerten (siehe Abbildung 4.2) ergeben, werden in Tabelle 4.2 dargestellt. Analog zu Modell 1 werden auch bei Modell2 die Abweichungen zwischen den empfangenen Signalen und dem jeweiligen angepassten Referenzsignal ausgewertet. Die größte Abweichung zwischen den Signalen ergibt sich an ECU 8, was folglich zum größten MSE-Wert ( $MSE = 0,112286$ ) im Modell führt.

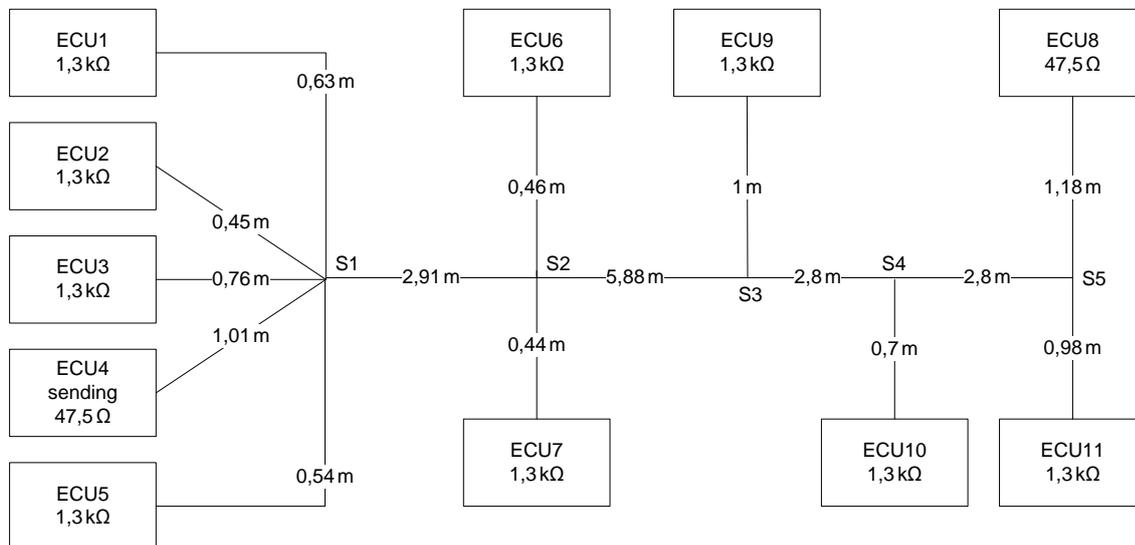


Abbildung 4.2: Topologie von Modell 2

Bus ECU	MSE
1	0,058914
2	0,059489
3	0,065180
4	0,028862
5	0,062439
6	0,106554
7	0,105730
8	0,112286
9	0,088132
10	0,087448
11	0,086482

Tabelle 4.2: Modell 2 - Fehlerwerte mit initialen Parameterwerten

## 4.2 Anpassen des Simulated Annealing Algorithmus

Der Algorithmus des Simulated Annealing wurde bereits in [Gü07] zur Optimierung innerhalb der Bussystemanalyse erfolgreich eingesetzt. Abgesehen vom reinen Einsatz des Optimierungsverfahren liefert Harald Günther in seiner Arbeit jedoch keine weiteren grundlegenden Informationen zum Verhalten des Optimierungsalgorithmus. Insbesondere bleibt Harald Günther den Vergleich zu anderen Algorithmen im Bezug auf Laufzeiten und erreichte Ergebnisgüte schuldig. Dies wird im Folgenden nachgeholt.

Die Durchführung der Analysen des Simulated Annealing Algorithmus dient dazu, eine Vergleichsbasis für die beiden neuen Algorithmen (GA, PSO) zu schaffen. Die Ergebnisse der Analyse sollen zeigen, ob und in welcher Form die populationsbasierten Ansätze des genetischen Algorithmus und der Partikel Schwarm Optimierung im vorliegenden Anwendungsfall Vorteile erzielen können. Weiterhin ist zu beurteilen, ob aufgrund der deutlich geringer zu erwartenden Zahl an Simulationen der Ansatz des Simulated Annealing letztendlich doch praktikabler ist. Da bereits eine entsprechende Implementierung des Verfahrens vorliegt, wird diese auch für die folgenden Analysen genutzt.

Bevor mit der eigentlichen Analyse begonnen wird, werden kurz die getroffenen Implementierungsentscheidungen sowie die initiale Wahl der Ablaufparameter erläutert und begründet.

### 4.2.1 Parameterauswahl und Variation

Da zwischen den einzelnen Variationsparametern des Bussystems keine unmittelbar nutzbaren mathematischen Abhängigkeiten existieren und außerdem die grundlegenden Fähigkeiten des Algorithmus auch im Hinblick auf Optimierungsaufgaben zu bewerten ist, über die kein oder nur sehr wenig Vorwissen existiert, wird die Auswahl des Parameters, dessen Wert zur Bestimmung einer FolgeLösung variiert werden soll, uniform zufällig durchgeführt. Der Wert des ausgewählten Parameters wird für die Berechnung einer FolgeLösung auf einen zufällig gewählten Wert innerhalb seines definierten Wertebereiches gesetzt. Somit soll eine gute Abdeckung des gesamten Suchraums bei der Optimierung der Parameterwerte gewährleistet werden. Alternativ wäre es möglich, den Wert des Parameters um einen festen prozentualen Anteil zu verändern. Ein solches Vorgehen erhöht aber die Wahrscheinlichkeit, dass die Suche nur in einem relativ kleinen Teil des Suchraums verläuft, da mehrfache Änderungen des selben Parameters in die selbe Richtung nötig wären, um die Suche in einen anderen Bereich des Suchraums zu verlagern.

### 4.2.2 Cooling-Schedule

Für die Implementierung wird ein einfaches Cooling-Schedule eingesetzt. Ausgehend von einem Kontrollparameter  $c = 1$  wird eine festgelegte Anzahl an FolgeLösungen berechnet,

die direkt abhängig von der Zahl der variablen Parameter des zu analysierenden Bussy-  
systems sind. Bei  $n$  variablen Parametern werden jeweils  $2n$  Folge-lösungen berechnet, ehe  
der Wert des Kontrollparameters dekrementiert wird. Die Dekrementierung des Kontroll-  
parameters erfolgt durch Multiplikation mit einer spezifizierten Variable  $\alpha$ . In [Gü07] wird  
für den Parameter der Wert  $\alpha = 0,8$  gewählt. Diese Wahl des Parameterwertes wird als  
Ausgangsbasis für die hier durchgeführten Analysen zunächst übernommen.

### 4.2.3 Wahl der Abbruchparameter

Der Algorithmus terminiert, sobald

- für den Kontrollparameter  $c$  ein Wert  $c < 0,1$  erreicht wird.
- der Algorithmus innerhalb von 30 % der berechneten Maximallaufzeit (gerechnet vom  
Zeitpunkt des Auffindens einer verbesserten Lösung) keine noch bessere Lösung fin-  
det.

### 4.2.4 Initiale Analyse des Simulated Annealing

Zu Beginn wird der Algorithmus mit den Ablaufparametern wie er bereits in [Gü07] ver-  
wendet wurde eingesetzt (siehe Abschnitt 4.2.2). Dieser Testlauf liefert die Vergleichsdaten,  
um den Nutzen der späteren Veränderungen der Ablaufparameter und der eingesetzten Er-  
weiterungen bewerten zu können.

#### 4.2.4.1 Modell 1

Um eine gewisse Absicherung der Ergebnisse zu erreichen, werden insgesamt zehn Ana-  
lyseläufe an Modell 1 durchgeführt. Eine detaillierte Auflistung der Ergebnisse der einzel-  
nen Analyseläufe findet sich im Anhang (siehe Tabelle A.1). Im Mittelwert erreichen die  
Analysen einen maxMSE-Wert von 0,728450 bei einer Standardabweichung von 0,003970  
(siehe Tabelle 4.3). Das entspricht einer Steigerung des maxMSE-Wertes um 29,24 % zum  
Referenzwert der initialen Parameterbelegung in Abschnitt 4.1.1. Dabei werden pro Bus-  
Analyse insgesamt 352 Simulationen durchgeführt. Keiner der durchgeführten Läufe ter-  
miniert vorzeitig.

<b>Mittelwert maxMSE</b>	0,728450
<b>Standardabweichung maxMSE</b>	0,003970
<b>Vergleich maxMSE</b>	129,24 %
<b>Maximum maxMSE</b>	0,734617
<b>Minimum maxMSE</b>	0,725755
<b>∅ Anzahl Simulationen</b>	352

**Tabelle 4.3:** SA - initiale Analyse - Modell 1

#### 4.2.4.2 Modell 2

Um eine eventuelle Abhängigkeit der Performance des Algorithmus von der Größe des Busmodells oder von der Anzahl der Variationsparameter des Modells aufzuzeigen, werden auch an Modell 2 identische Analysen durchgeführt. Um die späteren Anpassungen in ihrer Effektivität beurteilen zu können, muss jedoch zunächst die initiale Performance des Algorithmus bestimmt werden. Dazu wird wieder von den Ablaufparametern ausgegangen, die bereits zur initialen Analyse in Abschnitt 4.2.4.1 verwendet wurden. Analog zu den Tests an Modell 1 werden auch hier wieder zehn Analysedurchläufe ausgeführt. Die Ergebnisse aller einzelnen Analysen finden sich im Anhang dieser Arbeit (siehe Tabelle A.2). Die bei den zehn Analysen im Mittel erreichten Ergebniswerte werden in Tabelle 4.4 dargestellt. Der Mittelwert der bestimmten maxMSE-Werte der Analysen liegt bei 0,154123, dies entspricht einer Steigerung des maxMSE-Wertes um 37,39 % zum Referenzwert der initialen Parameterbelegung in Abschnitt 4.1.2. Die Standardabweichung der einzelnen Ergebnisse vom Mittelwert beträgt 0,001068. Alle Analysen führen die jeweils maximal möglichen 902 Simulationen während der Optimierung durch. Keiner der Optimierungsläufe terminiert vorzeitig.

<b>Mittelwert maxMSE</b>	0,154123
<b>Standardabweichung maxMSE</b>	0,001068
<b>Vergleich maxMSE</b>	137,37 %
<b>Maximum maxMSE</b>	0,155913
<b>Minimum maxMSE</b>	0,152287
<b>∅ Anzahl Simulationen</b>	902

**Tabelle 4.4:** SA - initiale Analyse - Modell 2

#### 4.2.5 Variation Cooling-Schedule

Der einfachste und schnellste Weg die Ergebnisse zu beeinflussen, die mit Hilfe eines Simulated Annealing Algorithmus erreicht werden, besteht darin, Änderungen am Cooling-Schedule vorzunehmen. In der hier gewählten Implementierung bietet sich für eine Beeinflussung des Algorithmus durch das Cooling-Schedule der Parameter  $\alpha$  an. Der Parameter legt die jeweilige Schrittweite fest, um die der Wert des Kontrollparameters  $c$ , also die „Temperatur“ im Algorithmus, reduziert wird. Die Dekrementierung des Kontrollparameters wird jeweils nach einer festgelegten Zahl berechneter Folge­lösungen durchgeführt. Dies hat sowohl direkten Einfluss auf die Laufzeit des Gesamtalgorithmus als auch auf die Akzeptanzwahrscheinlichkeit von schlechteren Lösungen (siehe Abschnitt 3.4). Somit sind direkte Auswirkungen auf die erzielten Ergebnisse zu erwarten. Eine Wahl des Parameterwertes für  $\alpha$  nahe Eins führt zu einem sehr langsamen Absinken des Kontrollparameters  $c$ . Somit wird eine große Anzahl an Simulationen während der Optimierung durchgeführt

bis der Wert des Kontrollparameters schließlich das Abbruchkriterium erfüllt. In der Folge ergibt sich eine deutlich höhere Zahl durchgeführter, nicht zwingend benötigter Simulationen, die nach Möglichkeit vermieden werden sollten. Bei einer hohen Zahl an Simulationen ist zwar zu erwarten, dass die erzielten Ergebnisse mit hoher Wahrscheinlichkeit vergleichsweise gut sind, weil aber die Simulation mit Abstand die zeitaufwändigste Komponente der kompletten Bus-Analyse ist, wäre die Laufzeit unverhältnismäßig hoch, bis diese Ergebnisse vorlägen. Ein zu schnelles Absinken des Kontrollparameters  $c$  ist ebenso zu vermeiden, da sich in diesem Falle die Suche zu schnell auf einen zu kleinen Suchbereich konzentrieren würde und deshalb mit großer Wahrscheinlichkeit nur ein lokales Optimum gefunden würde.

Aus den dargestellten Gründen ist der erste Arbeitsschritt bei der Anpassung des Simulated Annealing, den Einfluss des Parameters  $\alpha$  auf Laufzeiten und Ergebnisgüte zu bestimmen.

#### 4.2.5.1 Modell 1

Die initiale Belegung für den Wert des Parameters beträgt  $\alpha = 0,8$ . Ausgehend von diesem Wert wird sowohl die Auswirkung einer Steigerung und damit einer Erhöhung der möglichen Algorithmuslaufzeit sowie eine Verringerung des Parameterwertes und somit eine Verringerung der maximal möglichen Zahl an Simulationen getestet. Es werden für die Werte  $\alpha = 0,7$  und  $\alpha = 0,9$  wieder jeweils zehn Analysen des ersten Modells durchgeführt. Die detaillierten Ergebnisse zu den einzelnen Optimierungen befinden sich im Anhang dieser Arbeit (siehe Tabelle A.3 und A.4). Tabelle 4.5 stellt die im Mittel über die jeweils zehn durchgeführten Optimierungen erzielten Ergebnisse den bisher erreichten Ergebnissen gegenüber, die in Abschnitt 4.2.4.1 mit der initialen Wertebelegung des Parameters berechnet wurden. Es ist zu beobachten, dass die Ergebnisse bei einer langsameren Dekrementierung des Kontrollparameters erkennbar besser sind. Bei einer Erhöhung des Parameters  $\alpha = 0,8$  auf  $\alpha = 0,9$  steigt der erreichte maxMSE-Wert um 2%. Die Zahl der durchgeführten Simulationen verdoppelt sich im Vergleich dabei auf 704 Simulationen. Eine Verringerung des  $\alpha$ -Parameters auf  $\alpha = 0,7$  führt hingegen im Mittel, im Vergleich zu den Analysen mit  $\alpha = 0,8$ , zu keiner weiteren Verschlechterung der Ergebnisse. Allerdings streuen die Ergebnisse stärker um den berechneten Mittelwert. Die Varianz der erzielten Fehlerwerte verdreifacht sich nahezu. Bei Betrachtung der Konvergenzverläufe fällt auf, dass die Optimierungsläufe mit  $\alpha = 0,7$  und  $\alpha = 0,8$  ein nahezu identischen Konvergenzverlauf aufweisen, wie die Optimierungsläufe mit  $\alpha = 0,9$  (siehe Abbildung 4.3). Durch das schnelle Absinken des Kontrollparameters wird jedoch das Abbruchkriterium früher erfüllt, so dass die Endergebnisse der Läufe mit  $\alpha = 0,9$  nicht erreicht werden können.

$\alpha$	0,8	0,7	0,9
Mittelwert maxMSE	0,728450	0,725607	0,743580
Standardabweichung maxMSE	0,003970	0,011638	0,005450
Vergleich maxMSE	100,00 %	99,61 %	102,08 %
Maximum maxMSE	0,734617	0,739528	0,748165
Minimum maxMSE	0,725755	0,705645	0,731502
$\emptyset$ Anzahl Simulationen	352	224	704
Vergleich Sim-Anzahl	100,00 %	63,64 %	200,00 %

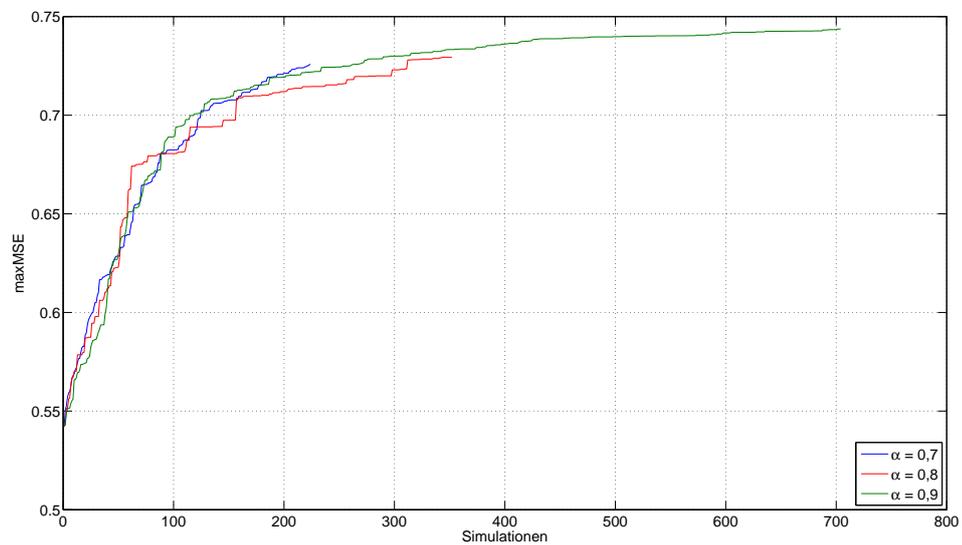
Tabelle 4.5: SA - Einfluss des Parameters  $\alpha$  - Modell 1

Abbildung 4.3: SA - Konvergenzverhalten - Modell 1

#### 4.2.5.2 Modell 2

An Modell 2 werden ebenfalls jeweils zehn Analysen mit den Parameterwerten  $\alpha = 0,7$  und  $\alpha = 0,9$  durchgeführt, um die Auswirkungen einer Steigerung oder einer Reduzierung der Zahl an Simulationen während der Optimierungen zu bestimmen. Alle Ergebnisse der einzelnen Analysen sind wieder im Anhang zu finden (siehe Tabelle A.5 und A.6). Bei der Betrachtung der durchschnittlich erreichten Fehlerwerte zeigen sich im Vergleich zu den Analysen des kleineren Modells einige Unterschiede. Bei Modell 1 erzeugten die Analysen mit  $\alpha = 0,8$  und  $\alpha = 0,7$  MSE-Werte, die lediglich um einen halben Prozentpunkt differierten. Bei Modell 2 ergibt sich eine etwas größere Differenz von etwas über 3% zwischen den durchschnittlich erreichten maxMSE-Werten. Bei  $\alpha = 0,9$  zeigen beide Analysen die besten Ergebnisse. Für Modell 2 liegen sie um etwa 1,4% über den Werten der Analysen mit einem Parameterwert  $\alpha = 0,8$  (siehe Tabelle 4.6). Auch die Streuung der Ergebniswerte kann bei langsamerer Abkühlung um ungefähr ein Viertel reduziert werden. Da wiederum keine der Analysen vorzeitig terminiert, verlängert sich auch die Laufzeit der Analysen durch die Verdoppelung der Simulationsanzahl (1804) auf etwa das Doppelte. An den gemittelten Konvergenzverläufen der Optimierungsläufe ist gut zu erkennen, dass die Konvergenzgeschwindigkeit der Läufe mit  $\alpha = 0,7$  und  $\alpha = 0,8$  nicht schlechter ist, als bei den Läufen mit  $\alpha = 0,9$  (siehe Abbildung 4.4). Allerdings reicht die Gesamtzahl der maximal möglichen Simulationen nicht aus, um am Ende das Ergebnisniveau der Läufe mit  $\alpha = 0,9$  zu erreichen.

$\alpha$	0,8	0,7	0,9
Mittelwert maxMSE	0,154123	0,149172	0,156314
Standardabweichung maxMSE	0,001068	0,004070	0,000774
Vergleich maxMSE	100,00%	96,79%	101,42%
Maximum maxMSE	0,155913	0,153859	0,157734
Minimum maxMSE	0,152287	0,144175	0,155309
∅ Anzahl Simulationen	902	574	1804
Vergleich Sim-Anzahl	100,00%	63,64%	200,00%

Tabelle 4.6: SA - Einfluss des Parameters  $\alpha$  - Modell 2

#### 4.2.5.3 Ergebnis

Die Ergebnisse der Optimierungsläufe an den beiden Testmodellen zeigen beide ähnliche Abhängigkeiten im Bezug auf die Geschwindigkeit mit der der Kontrollparameter abgesenkt wird. Für beide Modelle erzeugen jeweils die Optimierungsläufe die besten Ergebnisse, die mit dem größten Wert des Parameters  $\alpha$  durchgeführt wurden. Hierbei wird der Kontrollparameter am langsamsten abgesenkt. Dies führt in der Konsequenz allerdings zu einer deutlichen Steigerung der Simulationszahlen auf 200% im Vergleich zur initialen Analyse des Simulated Annealing in Abschnitt 4.2.4.

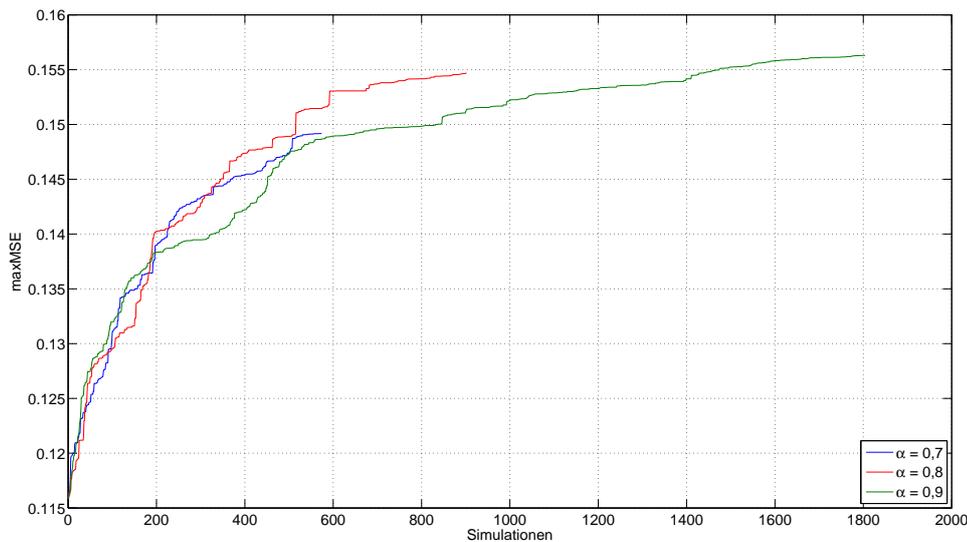


Abbildung 4.4: SA - Konvergenzverhalten - Modell 2

## 4.2.6 Lokalisierungsschritte

In Abschnitt 3.4.2 wurden Lokalisierungsschritte als eine Möglichkeit vorgestellt, die Performance der Optimierungen durch Simulated Annealing zu erhöhen. Die Technik der Lokalisierungsschritte wurde ebenfalls bereits in [Gü07] eingesetzt. Es fehlt allerdings bisher ein Nachweis für den durch sie erbrachten konkreten Nutzen in Bezug auf Laufzeit und Ergebnisgüte der durchgeführten Bussystemanalysen. Darum wird der Einfluss, den der Einsatz von Lokalisierungsschritten für die Parametervariation der Bussystemanalyse hat, im Folgenden untersucht.

### 4.2.6.1 Modell 1

Getestet wird, welchen Nutzen der Einsatz von Lokalisierungsschritten in Abhängigkeit der Wahl des Parameters  $\alpha$  erzielt. Zu diesem Zweck werden für alle drei der im obigen Abschnitt 4.2.5 getesteten Werte für  $\alpha$  jeweils zehn Bus-Analysen unter Verwendung von Lokalisierungsschritten durchgeführt. Die Schrittweite pro Lokalisierungsschritt beträgt dabei für alle Lokalisierungsschritte zunächst 2,5 % des aktuellen Parameterwertes. Die Ergebnisse der Einzelläufe finden sich im Anhang (siehe Tabelle A.7).

Auffällig ist die signifikante Verbesserung der Ergebnisse im Vergleich zu denen, die ohne den Einsatz von Lokalisierungsschritten erreicht wurden. Insbesondere wird dies bei Betrachtung der Analysen deutlich, bei denen der Wert des Parameters  $\alpha = 0,7$  beträgt. Die in ihnen erzielten Ergebnisse haben die gleiche Güte wie die Analysen ohne Lokalisierungs-

schritte mit einem Wert  $\alpha = 0,9$ . Dabei führen sie jedoch zu einer deutlich reduzierten Anzahl an Simulationen. Es können im Schnitt knapp 60 % der Simulationen eingespart werden, der resultierende Zeitgewinn ist erheblich.

Einfluss auf den Erfolg, der durch den Einsatz von Lokalisierungsschritten erreicht wird, hat mitunter auch die jeweilige Schrittweite der einzelnen Lokalisierungsschritte. Daher werden zusätzliche Analysen des Bussystems durchgeführt, bei denen die Schrittweite auf 5 % und 10 % erhöht wird. Die Ergebnisse der einzelnen Analyseläufe sind wieder im Anhang zu finden (siehe Tabellen A.8 und A.9).

Bei Betrachtung der Ergebnisse ist zu erkennen, dass durch den Einsatz einer größeren Schrittweite der Lokalisierungsschritte die Suchzeit zum Teil deutlich verkürzt werden kann. Im Mittel erreichen alle Analysen mit zunehmender Schrittweite in weniger Simulationen ein gleichbleibendes oder besseres Ergebnisniveau.

Die Tabelle 4.7 stellt die gemittelten Ergebnisse einander gegenüber, die durch Verwendung von Lokalisierungsschritten mit den unterschiedlichen Schrittweiten erreicht werden konnten. Als Referenz zum Vergleich dienen die besten Ergebnisse, die in Abschnitt 4.2.5 ohne den Einsatz von Lokalisierungsschritten erreicht wurden. Die Verbesserung in den Laufzeiten der Analysen ist dabei zum Teil beachtlich. So kann durch den Einsatz von Lokalisierungsschritten ohne Einbußen im Ergebnisniveau eine Einsparung von knapp 62 % der Simulationen erreicht werden.

$\alpha$	0,9		0,7		
Schrittweite	ohne	2,5 %	5 %	10 %	
Mittelwert maxMSE	0,743580	0,749086	0,750117	0,753457	
Standardabweichung maxMSE	0,005450	0,007211	0,004383	0,002004	
Vergleich maxMSE	100,00 %	100,74 %	100,88 %	101,33 %	
Maximum maxMSE	0,748165	0,754399	0,754607	0,754933	
Minimum maxMSE	0,731502	0,731334	0,743560	0,748203	
Anzahl Simulationen	704	295	275	269	
Vergleich Sim-Anzahl	100,00 %	41,90 %	39,02 %	38,18 %	

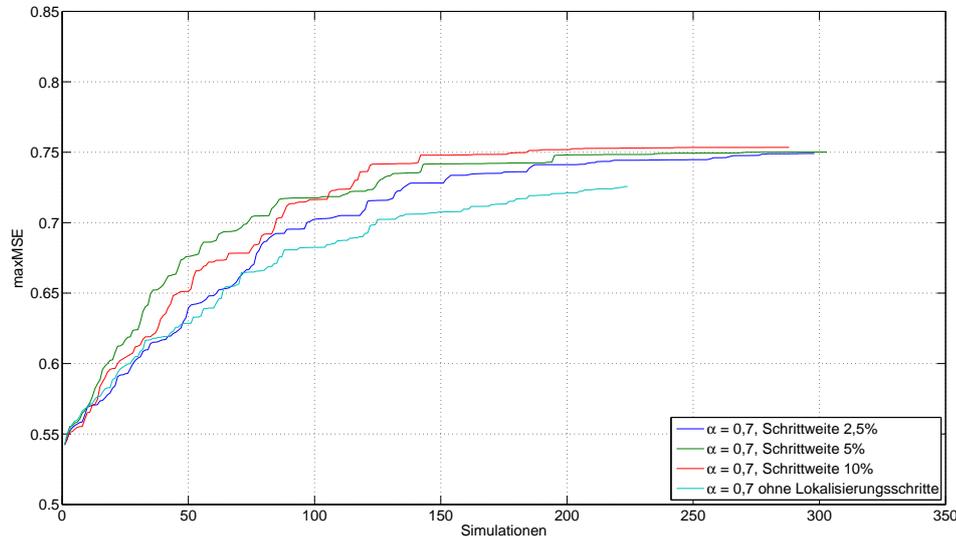
  

$\alpha$	0,8			0,9		
Schrittweite	2,5 %	5 %	10 %	2,5 %	5 %	10 %
Mittelwert maxMSE	0,751939	0,753108	0,750654	0,753939	0,753133	0,753221
Standardabweichung maxMSE	0,003871	0,001387	0,005412	0,000539	0,003214	0,001522
Vergleich maxMSE	101,12 %	101,28 %	100,95 %	101,39 %	101,28 %	101,30 %
Maximum maxMSE	0,755338	0,754930	0,755006	0,754593	0,754747	0,754652
Minimum maxMSE	0,741767	0,751135	0,740617	0,753111	0,744066	0,749814
∅ Anzahl Simulationen	403	401	383	733	748	653
Vergleich Sim-Anzahl	57,22 %	56,93 %	54,38 %	104,06 %	106,28 %	92,81 %

Tabelle 4.7: SA - Einfluss Lokalisierungsschritte - Modell 1

Bei der Betrachtung der Konvergenzverläufe zeigen sich für alle Optimierungsläufe erkennbar schneller ansteigende Verläufe als in den Optimierungsläufen ohne Lokalisierungsschritte. Exemplarisch werden die gemittelten Konvergenzverläufe für  $\alpha = 0,7$  in der Abbildung

4.5 dargestellt. Die gemittelten Konvergenzverläufe der übrigen Optimierungsläufe finden sich im Anhang dieser Arbeit (siehe Abbildung A.1 und A.2).



**Abbildung 4.5:** SA - Lokalisierungsschritte für  $\alpha = 0,7$ : Konvergenzverhalten - Modell 1

#### 4.2.6.2 Modell 2

Die Auswirkungen des Einsatzes von Lokalisierungsschritten werden auch an Modell 2 ausgewertet. Es werden, wie bei Modell 1, ebenfalls die Ergebnisse der Werte 0,7, 0,8, 0,9 für  $\alpha$  untersucht. Die anfängliche Schrittweite der Lokalisierungsschritte wird zunächst auf 2,5 % des aktuellen Parameterwertes gesetzt. Für jeden Wert des Parameters  $\alpha$  werden zunächst fünf Analysen durchgeführt. Die Ergebnisse aller Einzelanalysen finden sich im Anhang dieser Diplomarbeit (siehe Tabelle A.10).

Beim Vergleich der Ergebnisse untereinander dienen die besten Ergebnisse aus 4.2.5.2 als Referenz, um den Nutzen zu verdeutlichen, der durch den Einsatz der Lokalisierungsschritte erreicht wird. Es ist ersichtlich, dass der Einsatz der Lokalisierungsschritte den Algorithmus in die Lage versetzt, bereits bei Verwendung eines Wertes  $\alpha = 0,7$  eine Ergebnisgüte zu erzielen, die ohne Lokalisierungsschritte nur bei Verwendung von  $\alpha = 0,9$  zu erreichen war, bei jedoch einer deutlich höheren Simulationszahl während der Optimierungen. Wie bei Modell 1 können gegenüber den Analysen in 4.2.5.2 knapp 60 % der Simulationen eingespart werden. Im Schnitt benötigen die Analysen 757 Simulationen (zum Vergleich: 1804 durchgeführte Simulationen vorher). Eine Erhöhung des Parameters  $\alpha$ , also eine langsamere Abkühlung und folglich eine höhere Anzahl möglicher Simulationen im Optimierungsverlauf, verbessern die ermittelten Ergebniswerte nicht mehr. Auch die

Streuung der Ergebnisse wird nicht mehr verringert.

Als nächstes werden unterschiedlich große Schrittweiten analog zu Abschnitt 4.2.6.1 für die eingesetzten Lokalisierungsschritte getestet. Es wird wieder eine Erhöhung der Schrittweite auf 5 % bzw. auf 10 % vom aktuellen Wert des betrachteten Parameters in Kombination mit den drei bekannten  $\alpha$ -Werten über jeweils fünf Bus-Analysen untersucht. Die Ergebnisse der einzelnen Analysen sind im Anhang dieser Arbeit zu finden (siehe Tabellen A.11, A.12).

Durch eine Erhöhung der Schrittweite auf 5 % kann für  $\alpha = 0,7$  die Anzahl der benötigten Simulationen noch einmal um knapp 5 % gegenüber den Optimierungsläufen reduziert werden, die eine Schrittweite der Lokalisierungsschritte von 2,5 % verwendet haben. Für  $\alpha = 0,9$  unterscheiden sich die Ergebniswerte weder in den gefundenen maxMSE-Werten noch in der Anzahl der Simulationen signifikant. Tabelle 4.8 stellt die gemittelten Ergebnisse einander gegenüber, die bei Einsatz der Lokalisierungsschritte mit den unterschiedlichen Schrittweiten in den Analyseläufen des Busmodells erreicht werden konnten. Die Erhöhung der Schrittweite auf 10 % erreicht für die Optimierungsläufe mit  $\alpha = 0,7$  und  $\alpha = 0,9$  im Mittel eine geringfügige Verbesserung der gefundenen maxMSE-Werte. Allerdings steigt die Varianz der ermittelten Ergebnisse wieder etwas an. Eine Ausnahme bilden die Optimierungsläufe mit  $\alpha = 0,8$  und einer Lokalisierungsschrittweite von 10 %. Diese führen zu keiner weiteren Verbesserung der Ergebnisse gegenüber den Optimierungsläufen die eine Schrittweite von 5 % verwendet haben. Auch die Varianz der ermittelten maxMSE-Werte steigt nicht so stark wie bei den anderen Optimierungsläufen mit gleicher Schrittweite aber einem anderen Wert für  $\alpha$  an.

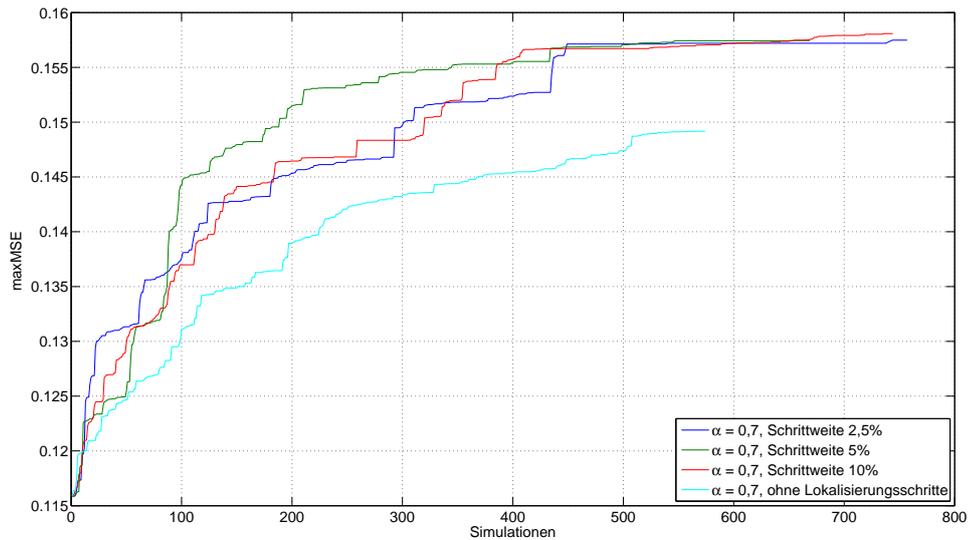
$\alpha$	0,9		0,7		
Schrittweite	ohne	2,50 %	5,00 %	10 %	
Mittelwert maxMSE	0,156314	0,157646	0,157573	0,158093	
Standardabweichung maxMSE	0,000774	0,000338	0,000472	0,001333	
Vergleich maxMSE	100,00 %	100,85 %	100,81 %	101,14 %	
Maximum maxMSE	0,157734	0,157884	0,158090	0,159798	
Minimum maxMSE	0,155309	0,157064	0,157226	0,156425	
∅ Anzahl Simulationen	1804	756,60	668,40	744,20	
Vergleich Sim-Anzahl	100,00 %	41,94 %	37,05 %	41,25 %	

$\alpha$	0,8			0,9		
Schrittweite	2,50 %	5,00 %	10 %	2,50 %	5,00 %	10 %
Mittelwert maxMSE	0,157785	0,156551	0,157225	0,157665	0,157639	0,158181
Standardabweichung maxMSE	0,000622	0,000687	0,000890	0,000937	0,000898	0,001813
Vergleich maxMSE	100,94 %	100,15 %	100,58 %	100,86 %	100,85 %	101,19 %
Maximum maxMSE	0,158655	0,157180	0,158191	0,159610	0,158463	0,159671
Minimum maxMSE	0,157196	0,155519	0,156381	0,156506	0,156506	0,155971
Anzahl Simulationen	1260,80	1040,00	928,40	1759,80	1733,20	2053,83
∅ Vergleich Sim-Anzahl	69,89 %	57,65 %	51,46 %	97,55 %	96,08 %	113,85 %

Tabelle 4.8: SA - Einfluss Lokalisierungsschritte - Modell 2

Die Verbesserung im Konvergenzverhalten die durch den Einsatz der Lokalisierungsschritte erreicht wird, ist besonders an den Konvergenzverläufen für  $\alpha = 0,7$  zu erkennen. Unabhängig von der Schrittweite ist der Konvergenzverlauf der Optimierungsläufe mit Lokalisierungsschritten deutlich steiler als ohne (siehe Abbildung 4.6). Die Konvergenzverläufe der Optimierungsläufe für  $\alpha = 0,8$  und  $\alpha = 0,9$  finden sich im Anhang dieser Arbeit (siehe Abbildung A.3 und A.4).



**Abbildung 4.6:** SA - Lokalisierungsschritte für  $\alpha = 0,7$ : Konvergenzverhalten - Modell 2

### 4.2.6.3 Ergebnis

Die Ergebnisse zeigen, dass sich durch den Einsatz von Lokalisierungsschritten die Performance des Simulated Annealing und damit die Performance der Bus-Analyse im vorliegenden Anwendungsfall deutlich verbessern lässt. Werden Lokalisierungsschritte eingesetzt, kann der Kontrollparameter deutlich schneller dekrementiert werden, was folglich zu einer Verringerung der Zahl an Simulationen führt, ohne dass die Ergebnisgüte darunter leidet. Im Gegenteil, mit Lokalisierungsschritten sind auch bei einem deutlich schnelleren Absinken des Kontrollparameters die Ergebnisse sogar minimal besser. Besonders hervorzuheben sind die Ergebnisse, die sich an Modell 1 für  $\alpha = 0,7$  erreichen lassen. Lagen die Ergebnisse ohne Lokalisierungsschritte schon bei  $\alpha = 0,8$  deutlich unter den Ergebnissen mit  $\alpha = 0,9$ , so lassen sich nun schon für  $\alpha = 0,7$  Ergebnisse erzielen, die besser sind als alle vorherigen und auf demselben Niveau liegen wie Ergebnisse, die mit Lokalisierungsschritten und höheren  $\alpha$ -Werten erreicht wurden. Beachtlich ist dabei, dass sich diese Ergebnisse trotz einer 60 %igen Einsparung an Simulationen erreichen lassen.

Interessant ist auch das Bild, dass sich bei Betrachtung der Ergebnisse in Abhängigkeit der unterschiedlichen Schrittweiten ergibt. So benötigen die Analysen, die eine Schrittweite von 10 % einsetzen, weniger Simulationen als die mit einer Schrittweite von 5 % oder 2,5 %. Hier wird insbesondere deutlich, welchen Vorteil eingesetztes Vorwissen für den Verlauf einer Optimierung bieten kann. Die Variationsparameter in den Busmodellen besitzen alle einen möglichen Variationsbereich von  $\pm 10\%$  ausgehend von ihrem Initialisierungswert. Da zu erwarten ist, dass die kritischen Parameterkonfigurationen eher in den Randbereichen der Wertebereiche zu finden sind, gelangen die Analysen, die eine größere Schrittweite für ihre Lokalisierungsschritte nutzen, schneller in die Nähe der Wertebereichsgrenzen und damit eher in die kritischen Bereiche. Allerdings darf die Schrittweite natürlich nicht zu groß gewählt werden. Bei einem Intervall von  $\pm 10\%$  würde es keinen Sinn machen, eine Schrittweite von 25 % einzusetzen. Bei einer zu großen Schrittlänge kann es insbesondere zu Problemen gegen Ende der Analyse kommen, wenn es darum geht, weitere, kleinere Verbesserungen zu finden. Als Alternative könnte eine zeitlich flexible Steuerung der Schrittweite eingesetzt werden, die ähnlich dem Kontrollparameter, dekrementiert wird. Zu Beginn der Analyse würde diese flexible Steuerung dazu führen, dass die Suche relativ schnell gute Suchbereiche erreicht. Durch die kleineren Schrittweiten wäre der Algorithmus gegen Ende der Analyse möglicherweise eher in der Lage Verbesserungen aufzuspüren, die recht nahe an den aktuellen Parameterwerten liegen.

### 4.3 Anpassen des genetischen Algorithmus

Im folgenden Abschnitt werden u. a. die Ablaufparameter des genetischen Algorithmus an die gegebene Problemstruktur angepasst. Dazu wird eine Reihe von Tests mit dem Ziel durchgeführt, den Einfluss verschiedener Werte der Ablaufparameter auf das Optimierungsergebnis zu ermitteln und die Anzahl der durchzuführenden Simulationen zu bestimmen. Insbesondere werden auch die unterschiedlichen, zur Auswahl stehenden Operatoren (siehe Abschnitt 3.6.4) des genetischen Algorithmus auf ihre Eignung für den Einsatz in der Kfz-Bussystemanalyse untersucht und getestet.

Für die Implementierung und das Einbinden eines genetischen Algorithmus in die bestehende Analyse-Umgebung wird die Matlab® Genetic Algorithm and Direct Search Toolbox™ eingesetzt<sup>1</sup>.

---

<sup>1</sup>Informationen unter <http://www.mathworks.com/products/gads/>

### 4.3.1 Abbruchkriterien

Für die Analysen der Bus-Modelle werden die Abbruchkriterien des genetischen Algorithmus wie folgt definiert:

- maximale Anzahl an Generationen:  $Generations = 100$
- Terminierung, wenn über eine festgelegte Anzahl *stall* von Generationen keine Verbesserung mehr gefunden wurde, mit  $stall = Generations \cdot 0,3$

### 4.3.2 Fitness-Skalierung

Aufgrund der in Abschnitt 3.6.5 erläuterten Probleme, die bei einer proportionalen Skalierung der Fitnesszuweisung auftreten können, kommt sie für die Berechnung der internen Fortpflanzungswahrscheinlichkeiten nicht in Frage. Insbesondere ist zu erwarten, dass die Zielfunktionswerte der einzelnen Individuen sehr nahe beieinander liegen und sich nur relativ gering unterscheiden. Deshalb wird die interne Zuweisung durch eine rangbasierte Skalierung erfolgen.

### 4.3.3 Elite-Übernahme

In der hier gewählten Implementierung des genetischen Algorithmus wird die Elite-Übernahme eingesetzt, die in Abschnitt 3.6.5 als Erweiterung des Standard-Algorithmus vorgestellt wurde. Durch das Verfahren der Elite-Übernahme soll sichergestellt werden, dass bereits gefundene, gute Lösungen beim randomisierten Auswahl- und Reproduktionsverfahren nicht wieder verloren gehen, sondern ihre genetischen Informationen auch in der nächsten Generation zur Erzeugung neuer Nachkommen zur Verfügung stehen. Damit aber die genetische Vielfalt in der Population möglichst wenig beeinträchtigt wird, werden nur die beiden besten Individuen direkt in die nächste Population übernommen.

### 4.3.4 Populationsgröße

Wie bereits in Abschnitt 3.6 dargestellt, handelt es sich bei genetischen Algorithmen um populationsbasierte Algorithmen. Es kann davon ausgegangen werden, dass die Größe der Population nicht nur direkten Einfluss auf die Anzahl der Simulationen hat, sondern auch auf die erzielten Ergebnisse. Daher wird der Einfluss unterschiedlicher Populationsgrößen auf die erzielte Güte der Ergebnisse und auf die resultierenden Laufzeiten der Optimierungen zuerst analysiert.

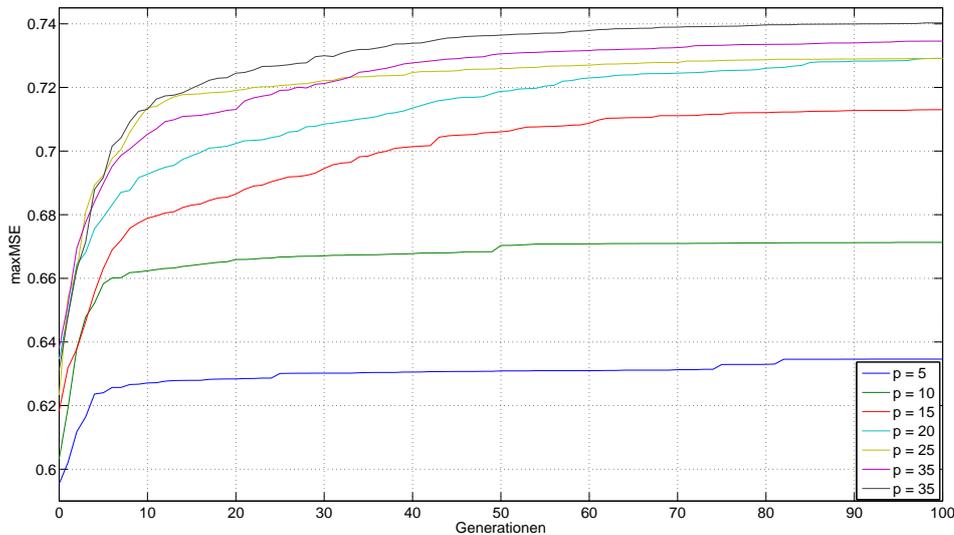
#### 4.3.4.1 Modell 1

Um eine geeignete Größe der Population für den genetischen Algorithmus zu finden, werden die Analysen des Bus-Modells mit unterschiedlichen Populationsgrößen  $p$  durchgeführt.

Dabei werden Werte für  $p = 5, 10, 15, 20, 25, 30, 35$  getestet. Für jeden Wert werden insgesamt zehn Analysen des Bus-Modells durchgeführt. Eine Auflistung aller Ergebnisse der einzelnen Analysen findet sich im Anhang dieser Arbeit (siehe Tabelle B.1).

Bei Betrachtung der Konvergenzverläufe, die in Abbildung 4.7 dargestellt werden und jeweils den durchschnittlichen Konvergenzverlauf über alle zehn durchgeführten Analysen pro Populationsgröße zeigen, ist deutlich zu erkennen, dass die bestimmten Fehlerwerte, die die Optimierungen erreichen, mit steigender Populationsgröße ebenfalls ansteigen.

Durch die größeren Populationen nimmt jedoch auch die Zahl der jeweils durchgeführten Simulationen stark zu. So führt im Schnitt eine Erhöhung der Populationsgröße von  $p = 5$  auf  $p = 20$  Individuen zu einer Steigerung der Zahl durchgeführter Simulationen um 368 %. Dabei wird eine durchschnittliche Verbesserung der Ergebnisse um knapp 15 % erreicht (siehe Tabelle 4.9). Optimierungen, bei denen die Populationsgröße  $p = 35$  Individuen beträgt, erreichen zwar nominell den besten Ergebniswert (im Vergleich zu  $p = 5$  eine Steigerung um 16,7 %), die Zahl der durchgeführten Simulationen steigt allerdings drastisch auf gut 830 % des Anfangswertes an, bis der Algorithmus schließlich terminiert. Die so erreichten Ergebnisse sind allerdings auch die stabilsten Ergebnisse mit der geringsten Abweichung vom Mittelwert.



**Abbildung 4.7:** GA - Populationsgröße: Konvergenzverhalten - Modell 1

#### 4.3.4.2 Modell 2

Ein identisches Bild zeigen die Analysen für Modell 2. Wie bei Modell 1 wurden Populationsgrößen von  $p = 5, 10, 15, 20, 25, 30, 35$  mit jeweils zehn Durchläufen pro Parameterwert

Populationsgröße	5	10	15	20
Mittelwert maxMSE	0,634603	0,671301	0,712996	0,729157
Standardabweichung maxMSE	0,024996	0,038325	0,025330	0,015440
Vergleich Ergebnisse	100,0 %	105,8 %	112,4 %	114,9 %
Maximum maxMSE	0,683353	0,725689	0,742096	0,744377
Minimum maxMSE	0,583424	0,618852	0,679714	0,692117
∅ Anzahl Simulationen	425,77	826,00	1482,00	1992,00
Vergleich Sim-Anzahl	100,00 %	194,00 %	348,08 %	467,86 %

Populationsgröße	25	30	35
Mittelwert maxMSE	0,729104	0,734556	0,740758
Standardabweichung maxMSE	0,009407	0,014404	0,007717
Vergleich Ergebnisse	114,9 %	115,8 %	116,7 %
Maximum maxMSE	0,741192	0,750743	0,750961
Minimum maxMSE	0,715450	0,709886	0,729326
∅ Anzahl Simulationen	2487,50	2977,50	3535,00
Vergleich Sim-Anzahl	584,24 %	699,32 %	830,26 %

Tabelle 4.9: GA - Populationsgröße - Modell 1

untersucht. Eine Darstellung aller Ergebnisse der einzelnen Analysen findet sich im Anhang in Tabelle B.2. Die gemittelten Konvergenzverläufe sind in Abbildung 4.8 zu sehen. Auch hier ist zu beobachten, dass grundsätzlich größere Populationen zu einer Verbesserung des Gesamtergebnisses führen. Durch die größeren Populationen ist aber erneut ein überproportionaler Anstieg durchgeführter Simulationen unumgänglich. Im Vergleich zu den Ergebnissen der Analysen an Modell 1 sind die Auswirkungen, die bei den Untersuchungen an Modell 2 beobachtet werden können, wie in Tabelle 4.10 zu sehen, noch einmal deutlicher. Der erreichte maxMSE-Wert für Populationen mit einer Größe von  $p = 20$  Individuen liegt gut 14 % über den Ergebnissen der Optimierungsläufe, die mit einer Populationsgröße  $p = 5$  erreicht werden. Hier zeigen beide Modelle nahezu identische Auswirkungen. Die Zahl der durchgeführten Simulationen bis zur Terminierung des Algorithmus liegt dabei im Vergleich bei 572 %. Die höchste getestete Populationsgröße  $p = 35$  erreicht auch in diesem Fall wieder das beste Ergebnis (18,4 % Verbesserung im Vergleich zu einer Populationsgröße  $p = 5$ ). Die Anzahl der durchgeführten Simulationen steigt dabei allerdings auf über 1000 % und damit auf fast das Doppelte zu den Analysen mit einer Populationsgröße  $p = 20$ .

#### 4.3.4.3 Ergebnis

Die Analysen an den beiden Testmodellen haben den Einfluss der unterschiedlichen Populationsgröße auf den Erfolg der Bus-Analysen im Hinblick auf die zu erwartende Laufzeit aufgezeigt. Grundsätzlich lässt sich daraus folgern, dass bei steigender Populationsgröße auch die erzielten Ergebnisse in ihrer Güte ansteigen. Allerdings steigt die Zahl der Simulationen bei ansteigender Populationsgröße überproportional an, so dass zwischen einer

Verbesserung der Ergebnisse und den deutlich verlängerten Optimierungslaufzeiten ausgewählt werden muss.

Populationsgröße	5	10	15	20
Mittelwert maxMSE	0,124741	0,132938	0,139351	0,142554
Standardabweichung maxMSE	0,008942	0,005393	0,002606	0,003483
Vergleich Ergebnisse	100,0 %	106,6 %	111,7 %	114,3 %
Maximum maxMSE	0,136894	0,138265	0,141490	0,152192
Minimum maxMSE	0,110566	0,119925	0,134448	0,138362
∅ Anzahl Simulationen	337,73	849,00	1242,27	1932,73
Vergleich Sim-Anzahl	100,00 %	251,4 %	367,8 %	572,3 %

Populationsgröße	25	30	35
Mittelwert maxMSE	0,143132	0,145812	0,147755
Standardabweichung maxMSE	0,002179	0,005991	0,004710
Vergleich Ergebnisse	114,7 %	116,9 %	118,4 %
Maximum maxMSE	0,146747	0,154640	0,152895
Minimum maxMSE	0,138324	0,139273	0,139386
∅ Anzahl Simulationen	2367,50	2865,00	3471,36
Vergleich Sim-Anzahl	701,0 %	848,3 %	1027,9 %

Tabelle 4.10: GA - Populationsgröße - Modell 2

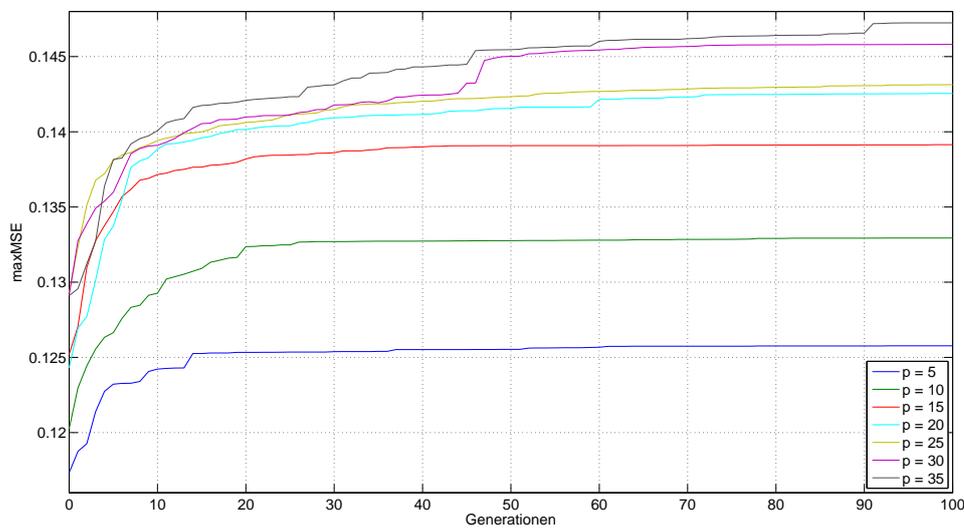


Abbildung 4.8: GA - Populationsgröße: Konvergenzverhalten - Modell 2

Bei beiden Modellen konnten die größten Ergebnisverbesserungen bei einer Populationsgröße  $p = 20$  erreicht werden. Die weiteren Verbesserungen, die durch eine Vergrößerung

der Populationen über  $p = 20$  erwirkt werden konnten, waren im Vergleich zum resultierenden Anstieg der Simulationszahlen nur sehr gering. Aus diesem Grund wird eine Populationsgröße von  $p = 20$  als guter Kompromiss zwischen guten Ergebnissen und einer möglichst geringen Zahl an Simulationen betrachtet. Für den weiteren Anpassungsprozess des genetischen Algorithmus wird deshalb eine Populationsgröße von  $p = 20$  eingesetzt.

### 4.3.5 Wahl des Selektions- und Crossover-Operators

In Abschnitt 3.6.4 wurden bereits eine Reihe unterschiedlicher Operatoren und Verfahren vorgestellt, mit denen zum einen die Selektion der Individuen und anschließend die Erzeugung neuer Nachkommen durchgeführt werden kann. Da die vorgestellten Operatoren und Selektionsverfahren zum Teil sehr unterschiedlich wirken, muss ihre konkrete Eignung für die jeweilige praktische Anwendung verifiziert werden. Dies geschieht im folgenden Abschnitt.

Verschiedene Operatoren werden die Ergebnisse der Optimierung direkt beeinflussen, weil Auswirkungen im Verlauf des Bus-Analysevorganges durch die unterschiedliche Lage der neu erzeugten Lösungen in Relation zu den alten Lösungen zu erwarten sind.

Darum werden im Folgenden verschiedene Tests durchgeführt, die die Eignung der verschiedenen Selektions- und Crossoververfahren aufdecken sollen.

Da mit einer gegenseitigen Beeinflussung zwischen Selektions- und Crossover-Operatoren zu rechnen ist, wird jeweils das Zusammenspiel der einzelnen Selektionsverfahren mit den unterschiedlichen Crossover-Operatoren näher betrachtet.

#### 4.3.5.1 Modell 1

Das erste getestete Selektionsverfahren ist die stochastisch uniforme Selektion. In Kombination mit jedem der zur Verfügung stehenden Crossover-Operatoren werden zunächst jeweils fünf Bus-Analysen durchgeführt. Die im Durchschnitt über jeweils fünf Analysen erzielten Ergebnisse der einzelnen Optimierungen zeigt die Tabelle 4.11. Bei der Auswertung fallen insbesondere die Ergebnisse negativ auf, die unter Verwendung des arithmetischen und des intermediären Crossover-Operators erzielt wurden. Die Ergebnisse beider Operatoren liegen recht deutlich, im Schnitt um etwa 7,5 %, unter den übrigen Ergebnissen. Eine Auslistung der Ergebnisse aller einzelnen Optimierungsläufe findet sich im Anhang in Tabelle B.3.

Wie in Abschnitt 3.6.4 dargestellt, bietet der intermediäre Crossover durch einen Parameter *Ratio* die Möglichkeit Einfluss auf sein Reproduktionsverhalten zu nehmen. Deshalb werden weitere Tests mit unterschiedlichen Werten für diesen Parameter durchgeführt, um auszuschließen, dass der ausgewählte Wert  $r = 0,5$  des Parameters zufällig besonders schlecht ausgewählt wurde. Es werden dabei die Werte  $r = 0,7, 0,8, 1,2, 1,5, 2$  für den Parameter getestet. Für jeden der Werte werden wieder insgesamt fünf Optimierungsläufe

Crossover-Operator	binär	1Punkt	2Punkt
Mittelwert maxMSE	0,729157	0,724469	0,726526
Standardabweichung maxMSE	0,015440	0,013019	0,006829
Vergleich Ergebnisse	100,0 %	99,4 %	99,6 %
Maximum maxMSE	0,744377	0,742464	0,734043
Minimum maxMSE	0,692117	0,710273	0,715627
∅ Anzahl Simulationen	1992,00	1893,33	2020,00
Vergleich Sim-Anzahl	100,00 %	95,05 %	101,41 %

Crossover-Operator	arithmetisch	heuristisch	intermediär
Mittelwert maxMSE	0,672915	0,733968	0,678902
Standardabweichung maxMSE	0,048398	0,015006	0,020011
Vergleich Ergebnisse	92,3 %	100,7 %	93,1 %
Maximum maxMSE	0,714369	0,748795	0,701097
Minimum maxMSE	0,618893	0,712746	0,654296
∅ Anzahl Simulationen	1836,00	2020,00	2020,00
Vergleich Sim-Anzahl	92,17 %	101,41 %	101,41 %

Tabelle 4.11: GA - stochastisch uniforme Selektion - Modell 1

durchgeführt, um den Einfluss des Parameters auf die Ergebnisse der Optimierung zu bestimmen. Tabelle 4.12 zeigt die Ergebnisse dieser zusätzlichen Tests. Als Vergleichsreferenz dienen die zuvor erzielten Ergebnisse mit  $r = 0,5$ . Es fällt auf, dass sich die Ergebnisse für Parameterwerte  $r \leq 1,2$  nicht signifikant ändern. Erst durch eine Erhöhung des Parameters auf Werte  $r \geq 1,5$  lassen sich annähernd Ergebnisse erzielen, die auf dem Niveau der Ergebnisse der anderen Crossoververfahren liegen.

Ratio	0,5	0,7	0,8	1,2	1,5	2
Mittelwert maxMSE	0,674463	0,683475	0,681606	0,674957	0,719439	0,723523
Standardabweichung maxMSE	0,018782	0,023995	0,014886	0,029610	0,021697	0,016924
Vergleich maxMSE	100,00 %	101,34 %	101,06 %	100,07 %	106,67 %	107,27 %
Maximum maxMSE	0,674463	0,707723	0,702975	0,716173	0,734781	0,737501
Minimum maxMSE	0,654296	0,650127	0,667497	0,637043	0,704097	0,696169
∅ Anzahl Simulationen	2020	2020	2020	2020	2020,00	2020,00
Vergleich Sim-Anzahl	100,00 %	100,00 %	100,00 %	100,00 %	100,00 %	100,00 %

Tabelle 4.12: GA - intermediärer Crossover: Einfluss des Ratio-Parameters - Modell 1

Die beschriebenen Ergebnisse führen zu dem Schluss, dass sich vor allem der arithmetische Crossover-Operator, aber auch der intermediäre, nicht für den Einsatz zur Parametervariation innerhalb der Bussystemanalyse eignet. Bei Betrachtung der Berechnungsgrundlagen zur Generierung der Nachkommen durch die jeweiligen Operatoren, drängt sich die Vermutung auf, dass das beobachtete Verhalten aus der Lage der erzeugten Nachkommen in Relation zu den Eltern resultiert. Sowohl beim intermediären, als auch beim arithmetischen Crossover, liegen die Nachkommen zwischen den Eltern, genauer gesagt innerhalb des durch die Eltern aufgespannten Hyperwürfels. Bei allen anderen Crossoververfahren liegen die Nachkommen außerhalb (heuristischer Crossover) oder auf den Eckpunkten des

Hyperwürfels (binärer, 1- und 2-Punkt Crossover). Offensichtlich sind die Positionen im Innern des Würfels für eine Suche nach dem globalen Optimum eine schlechte Wahl. Diese These wird von der Tatsache untermauert, dass die Ergebnisse bei einer deutlichen Steigerung des Ratio-Parameters innerhalb des intermediären Crossover-Operators signifikant steigen. Die Nachkommen liegen dann in diesem Falle mit einer größeren Wahrscheinlichkeit außerhalb des Hyperwürfels.

Es besteht allerdings die Möglichkeit, dass die Entwicklung der Suche lediglich langsamer verläuft und die Abbruchbedingungen für eine Suche mittels intermediärem oder arithmetischem Crossover relaxiert werden müssten, um auf ein zu den übrigen Verfahren vergleichbares Niveau zu gelangen. Diese Überprüfung wurde nicht mehr durchgeführt, da auch die benötigte Laufzeit bis zum Auffinden der besten Lösung ein Ziel der Anpassung des Algorithmus ist.

Auf Grund der beschriebenen Ergebnisse werden im Folgenden sowohl der arithmetische, als auch der intermediäre Crossover nicht weiter in die Analysen des Algorithmus einbezogen. Die übrigen Verfahren, die alle ein ähnlich gutes Niveau erreichen, werden im Zusammenwirken mit den weiteren Selektionsverfahren weiter getestet. Die Ergebnisse dieser Analysen sind nach den einzelnen Selektionsverfahren sortiert in Tabelle 4.13 abgebildet. Die starke Ähnlichkeit der Ergebnisse ist auffallend.

Für alle Selektionsverfahren erreichen die heuristischen Crossover-Operatoren im Mittel die besten Ergebnisse. Auch in der Anzahl der durchgeführten Simulationen innerhalb eines Analysedurchlaufes unterscheiden sich die Verfahren nur marginal. Nahezu alle Analysen terminieren erst durch Erreichen der maximal möglichen Anzahl an Generationen.

Während die ermittelten MSE-Fehlerwerte sich nur gering unterscheiden, zeigen sich in den Konvergenzverläufen der Optimierungen in Abhängigkeit vom verwendeten Crossover-Verfahren recht deutliche Unterschiede. Die Optimierungen mittels heuristischem Crossover zeigen einen eher langsamen, aber über den gesamten Verlauf relativ gleichmäßigen Anstieg. Die Verläufe der stochastischen sowie der 1- und 2-Punkt Operatoren steigen dagegen zunächst deutlich steiler an, verflachen mit Fortschreiten der Analysen aber zunehmend. Exemplarisch werden die Konvergenzverläufe der eingesetzten Crossover-Operatoren in Verbindung mit der Turnier-Selektion in Abbildung 4.9 gezeigt. Die graphische Darstellung der Konvergenzverläufe der übrigen Selektionsverfahren findet sich im Anhang (siehe Abbildungen B.1, B.2, B.3). Auch das gezeigte Konvergenzverhalten kann auf die räumliche Lage der Nachkommen in Relation zu ihren Eltern zurückgeführt werden. Während die Nachkommen, die durch den heuristischen Crossover-Operator erzeugt werden, in der Nähe des Elternteil mit dem besseren Fitnesswert, außerhalb des durch die Eltern aufgespannten Hyperwürfels liegen, befinden sich die Nachkommen, die durch die übrigen Operatoren erzeugt werden, stets auf den Eckpunkten des Würfels. Der heuristische Crossover-Operator berechnet intern die Position der Nachkommen mit einem Standard-Wert von 1,2. Somit ist der Abstand der Nachkommen, die durch den heuristischen Crossover-Operator erzeugt

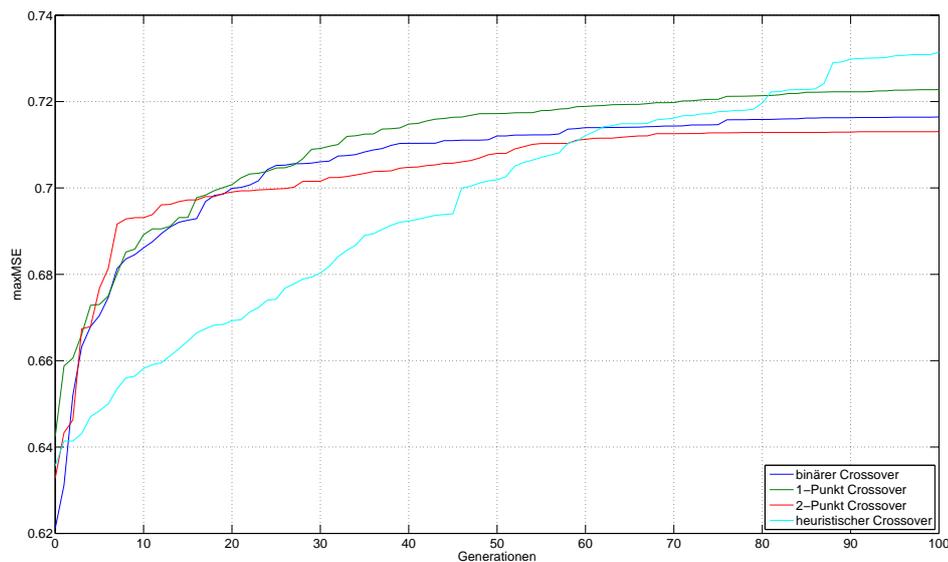
werden, zum Elter mit der besseren Fitness, deutlich geringer als bei den übrigen Verfahren.

Selektion	Roulette-Selektion			
Crossover	binär	1Punkt	2Punkt	heuristisch(1,2)
Mittelwert maxMSE	0,722516	0,707279	0,711609	0,731295
Standardabweichung maxMSE	0,009377	0,013668	0,018465	0,011614
Vergleich maxMSE	100,00 %	97,89 %	98,49 %	101,22 %
Maximum maxMSE	0,731914	0,720178	0,725684	0,731295
Minimum maxMSE	0,708985	0,682346	0,682246	0,717443
∅ Anzahl Simulationen	2020	1987	2016	2020
Vergleich Sim-Anzahl	100,00 %	98,35 %	99,80 %	100,00 %
Selektion	Turnier-Selektion			
Crossover	binär	1Punkt	2Punkt	heuristisch(1,2)
Mittelwert maxMSE	0,716440	0,722765	0,713037	0,731422
Standardabweichung maxMSE	0,021152	0,008219	0,019932	0,007834
Vergleich maxMSE	100,00 %	100,88 %	99,53 %	102,09 %
Maximum maxMSE	0,743766	0,736586	0,734123	0,740046
Minimum maxMSE	0,691796	0,714887	0,685499	0,722240
∅ Anzahl Simulationen	1900	2012	2008	2020
Vergleich Sim-Anzahl	100,00 %	105,89 %	105,68 %	106,32 %
Selektion	Rest-Selektion			
Crossover	binär	1Punkt	2Punkt	heuristisch(1,2)
Mittelwert maxMSE	0,726251	0,721677	0,718634	0,737456
Standardabweichung maxMSE	0,017833	0,014258	0,019584	0,010801
Vergleich maxMSE	100,00 %	99,37 %	98,95 %	101,54 %
Maximum maxMSE	0,741604	0,736420	0,739581	0,737456
Minimum maxMSE	0,697559	0,697948	0,689694	0,724248
∅ Anzahl Simulationen	1984	2020	2020	2020
Vergleich Sim-Anzahl	100,00 %	101,81 %	101,81 %	101,81 %

**Tabelle 4.13:** GA - Selektions- und Crossover-Verfahren - Modell 1

Ein einfaches Beispiel verdeutlicht den beschriebenen Sachverhalt:

Angenommen in einem gegebenen drei-dimensionalen Suchraum repräsentiert Elter 1 den Punkt  $m = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$  und Elter 2 den Punkt  $n = \begin{pmatrix} 2 \\ 2 \\ 2 \end{pmatrix}$ , wobei Elter 2 den höheren Fitnesswert besitzt. Dann ergibt sich als Nachkomme beider Eltern, bei Verwendung des heuristischen Crossovers (siehe Gleichung 3.4, Abschnitt 3.6.4.2) mit Default-Wert des Ratio-Parameters, der Punkt  $o = \begin{pmatrix} 2,2 \\ 2,2 \\ 2,2 \end{pmatrix}$ . Der Abstand zwischen den Punkten  $n$  und  $o$  beträgt demnach  $d_1 = 0,346$ . Werden die Nachkommen mit einem der anderen Crossover-Verfahren bestimmt, so beträgt ihr Abstand zum Elter mit der besseren Fitness  $d_2 \geq 1$ ,



**Abbildung 4.9:** GA - Turnierselktion: Konvergenzverhalten - Modell 1

je nachdem wie viele Parameter in der Lösung im Vergleich zum Elter variiert werden. Auf diese Weise ergibt sich ein größerer Abstand der Nachkommen zu ihren Eltern. Bei der Erzeugung der Nachkommen werden also größere Strecken im Suchraum zurückgelegt. Offensichtlich führt dies zu einer schnelleren Konvergenz der Suche, da zu Beginn größere Parameterwertebereiche übersprungen werden können.

#### 4.3.5.2 Modell 2

Zusätzlich zu den an Modell 1 durchgeführten Analysen werden weitere Analysen an Modell 2 durchgeführt. Unterschiede zwischen den Selektionsverfahren sollten sich aufgrund der gesteigerten Komplexität des Modells eher erkennen lassen. Dieses gilt natürlich auch für die Unterschiede der Crossover-Operatoren.

Aufgrund der bereits am vergleichsweise kleinen Bus-Modell gezeigten schlechten Ergebnisse des intermediären und des arithmetischen Crossovers wird auf eine weitere Untersuchung dieser beiden Operatoren am komplexeren Modell 2 verzichtet. Für die übrigen Crossover-Operatoren wird überprüft, ob sich für die Verfahren identische Verhaltensweisen zeigen, obwohl sich die Modellkomplexität gegenüber Modell 1 gesteigert hat. Dazu werden für alle möglichen Selektions-Crossover-Kombinationen jeweils drei Optimierungsdurchläufe ausgeführt.

Es zeigt sich im Bezug auf die erreichten Ergebnisse ein anderes Bild als bei Modell 1. Im ersten Modell lieferte jeweils der heuristische Crossover bei allen Selektionsverfahren die besten Ergebnisse. Dies ist bei Modell 2 nicht so. Hier werden die besten Ergebniswerte durch den binären Crossover-Operator erzeugt. Alle anderen Operatoren erreichen eine geringere

Ergebnisgüte. Einzig die Turnier-Selektion kann zusammen mit dem 2-Punkt Crossover minimal bessere Ergebnisse erreichen. In Bezug auf die Zahl der Simulationen erzielen die Rest-Selektion und die Turnier-Selektion gute Werte, bei annähernd gleichbleibender Ergebnisgüte. Unter den Crossover-Operatoren erzielen die 1- und 2-Punkt Crossover gute Ergebnisse, mit zum Teil deutlich geringeren Simulationszahlen, zusammen mit drei der getesteten Selektionsverfahren. Ausnahme ist hier die Roulette-Selektion. Tabelle 4.14 stellt die im Mittel erreichten Ergebnisse aus den jeweils verwendeten Selektions-Crossover-Kombinationen einander gegenüber.

Selektion	stochastisch uniforme Selektion			
Crossover	binär	1-Punkt	2-Punkt	heuristisch(1,2)
Mittelwert maxMSE	0,142554	0,137620	0,140496	0,138724
Standardabweichung maxMSE	0,003483	0,004442	0,002849	0,008462
Vergleich maxMSE	100,00 %	96,54 %	98,56 %	97,31 %
Maximum maxMSE	0,152192	0,142410	0,143785	0,147413
Minimum maxMSE	0,138362	0,133635	0,138851	0,130508
∅ Anzahl Simulationen	1932,73	1873,33	1633,33	1906,67
Vergleich Sim-Anzahl	100,00 %	96,93 %	84,51 %	98,65 %
Selektion	Roulette-Selektion			
Crossover	binär	1-Punkt	2-Punkt	heuristisch(1,2)
Mittelwert maxMSE	0,138497	0,134225	0,135674	0,137432
Standardabweichung maxMSE	0,002580	0,005221	0,006777	0,004782
Vergleich maxMSE	100,00 %	96,92 %	97,96 %	99,23 %
Maximum maxMSE	0,140403	0,139951	0,139955	0,140973
Minimum maxMSE	0,135561	0,129728	0,127860	0,131992
∅ Anzahl Simulationen	1620,00	1866,67	1780,00	1926,67
Vergleich Sim-Anzahl	100,00 %	115,23 %	109,88 %	118,93 %
Selektion	Turnier-Selektion			
Crossover	binär	1-Punkt	2-Punkt	heuristisch(1,2)
Mittelwert maxMSE	0,141317	0,134219	0,141998	0,136829
Standardabweichung maxMSE	0,000093	0,004829	0,001014	0,003813
Vergleich maxMSE	100,00 %	94,98 %	100,48 %	96,82 %
Maximum maxMSE	0,141383	0,139752	0,142584	0,141181
Minimum maxMSE	0,141251	0,130853	0,140827	0,134077
∅ Anzahl Simulationen	2020,00	1400,00	1446,67	1773,33
Vergleich Sim-Anzahl	100,00 %	69,31 %	71,62 %	87,79 %
Selektion	Rest-Selektion			
Crossover	binär	1-Punkt	2-Punkt	heuristisch(1,2)
Mittelwert maxMSE	0,140931	0,139512	0,140839	0,131186
Standardabweichung maxMSE	0,001371	0,002057	0,000893	0,009270
Vergleich maxMSE	100,00 %	99,43 %	100,37 %	93,49 %
Maximum maxMSE	0,141852	0,140796	0,142177	0,141698
Minimum maxMSE	0,139355	0,137140	0,140329	0,124184
∅ Anzahl Simulationen	2020,00	1726,67	1675,00	1666,67
Vergleich Sim-Anzahl	100,00 %	85,48 %	82,92 %	82,51 %

Tabelle 4.14: GA - Selektions- und Crossover-Verfahren - Modell 2

### 4.3.5.3 Ergebnis

Aus den Ergebniswerten und den Konvergenzverläufen der Optimierungsläufe an Modell 1 konnte zunächst angenommen werden, dass der heuristische Crossover-Operator zwar ein relativ langsames Konvergenzverhalten besitzt, aber in der Lage ist, die besten Ergebniswerte zu erzielen. Die anderen Crossoververfahren schienen geeignet zu sein, eine besonders schnelle Konvergenz zu erreichen, um beispielsweise bei der Analyse eines Bussystems eine erste grobe Einschätzung treffen zu können.

Diese Annahmen wurden durch die Tests an Modell 2 nicht bestätigt. Dort zeigte der heuristische Crossover-Operator nicht die besten Ergebnisse. Die besten Ergebnisse wurden mittels der stochastisch uniformen Selektion in Verbindung mit dem binären Crossover erreicht. Möglicherweise können die Ergebnisse des heuristischen Crossover-Operators noch verbessert werden, indem der *Ratio* Parameterwert vergrößert wird, so dass größere Schritte innerhalb des Suchraums möglich sind. Aufgrund der erzielten Ergebnisse kann jedoch keine klare Aussage getroffen werden, welcher der Operatoren sich am besten für den vorliegenden Anwendungsfall eignet. Insbesondere müssten die Werte aus den Optimierungsläufen an Modell 2 durch zusätzliche Läufe abgesichert werden, da sich in den bisherigen Läufen ein anderes Bild als bei Modell 1 ergibt.

### 4.3.6 Verhältnis zwischen Crossover- und Mutationsrate

Die Nachkommen einer Generation werden auf zwei unterschiedliche Arten aus den Individuen der aktuellen Generation gebildet. Der eine Teil wird von Crossoververfahren durch eine Vermischung der Eigenschaften mehrerer Individuen generiert. Zusätzlich wird ein zweiter Teil an Individuen geschaffen, die nur von jeweils einem Individuum abstammen. Hier werden die Eigenschaften der neuen Individuen durch sogenannte Mutationen, also zufällige Veränderungen einzelner Eigenschaften eines Individuums gebildet. Diese Mutationen halten die genetische Vielfalt innerhalb der Population aufrecht beziehungsweise fügen der Population neue genetische Informationen hinzu. Das Verhältnis zwischen den aus beiden Ansätzen erzeugten Individuen kann nachhaltigen Einfluss auf die Ergebnisse der Optimierungen haben. Ein sehr geringer Anteil an Mutationen führt dazu, dass die Individuen sich genetisch sehr schnell ähneln. In der Folge wird nur noch ein sehr kleiner Bereich des Suchraumes abgedeckt und ergo wird nur in diesem Teilbereich weiter gesucht. Ein Auffinden des globalen Optimums wird so erschwert und der Algorithmus konvergiert mit großer Wahrscheinlichkeit vorzeitig in einem lokalem Optimum. Ist allerdings der Anteil der Population zu groß, der durch Mutation erzeugt wird, führt dies dazu, dass die Suche mehr oder weniger einer Zufallssuche gleicht. Werden zu wenige neue Individuen auf der Basis guter alter Individuen generiert, kann die Suche kaum in Richtung einer Verbesserung der repräsentierten Lösungen konvergieren, da durch den hohen Anteil zufälliger Variationen die Suche zu stark streut.

#### 4.3.6.1 Modell 1

Im kleineren Testmodell werden die Auswirkungen unterschiedlicher Verhältnisse zwischen den beiden Reproduktionsverfahren getestet. Pro Verhältnis werden jeweils zehn Analysen durchgeführt. Dabei werden die folgenden Werte für die jeweiligen Reproduktionsanteile getestet:  $Crossover/Mutation = 80/20; 100/0; 60/40; 40/60; 20/80$ . Aufgrund der guten Ergebnisse in Abschnitt 4.3.5.1 wird für diese Tests als Selektionsverfahren die stochastisch uniforme Selektion ausgewählt und als Crossover-Operator der heuristische Crossover eingesetzt. Die Ergebnisse aller einzelnen Optimierungsläufe finden sich im Anhang dieser Arbeit in Tabelle B.4.

Beim Vergleich über die jeweils zehn durchgeführten Analysen zeigen sich beim gebildeten Mittel der erreichten maxMSE-Werte nur sehr geringe Unterschiede. Einzig die Analysen, die komplett auf Mutation verzichten, erzielen merklich schlechtere Ergebnisse. Auch in der Zahl der durchgeführten Simulationen unterscheiden sich die Auswirkungen der verschiedenen Verhältnisse nur marginal. Einzige Ausnahme bilden hier erneut die Analysen, die ihre Individuen einzig durch Crossover-Operatoren erzeugen. Tabelle 4.15 zeigt die über die jeweils zehn durchgeführten Analysen im Mittel erreichten Ergebnisse und die Zahl der durchgeführten Simulationen. Als Referenzwert wird das Crossover-Mutations-Verhältnis von 80/20 verwendet, das auch in den bisherigen Anpassungsschritten benutzt wurde.

Etwas deutlicher fallen die Unterschiede zwischen den Analysen bei Betrachtung der zugehörigen Konvergenzverläufe (siehe Abbildung 4.10) aus. Es fällt auf, dass die Analysen mit einem Crossoveranteil von 60 % und 80 % sowie die Analysen mit Crossoveranteilen von 20 % und 40 % fast identische Konvergenzverläufe zeigen. Interessant ist auch, dass die Konvergenzverläufe, in denen der größere Teil der Individuen einer Population durch Mutation erzeugt wird, zu jedem Zeitpunkt der Analyse des Bus-Modells die besseren Ergebnisse liefern.

Neben einem statisch festgelegten Verhältnis zwischen Crossover und Mutation kann das Verhältnis auch dynamisch ausgelegt werden. Der Algorithmus terminiert, wenn er über eine Spanne von 30 % seiner maximal möglichen 100 Generationen keine Verbesserung mehr gefunden hat (siehe Abschnitt 4.3.1). Bei der dynamischen Realisierung des Crossover-Mutations-Verhältnisses soll die Mutationsrate zu einem bestimmten Zeitpunkt kurzzeitig erhöht werden, um die genetische Vielfalt der Population wieder zu steigern. Der geeignete Zeitpunkt ist erreicht, kurz bevor das Abbruchkriterium erreicht wird. Dann wird kurzfristig die Mutationsrate gesteigert. Anfangs wird in den Tests ein fester Crossover-Anteil von 80 % neu erzeugter Individuen eingestellt. Sind 80 % des Abbruchkriteriums erreicht, wird das Verhältnis zwischen Crossover und Mutation für den nächsten Generationsschritt invertiert und anschließend wieder auf den vorherigen Wert gestellt. Sowohl diese dynamische Realisierung als auch ein früheres Invertieren des Verhältnisses nach 60 % des erreichten Abbruchkriteriums brachte keinen Erfolg.

Crossover:Mutation	0,8:0,2	1:0	0,6:0,4	0,4:0,6	0,2:0,8
Mittelwert MSE	0,734320	0,654153	0,733488	0,740720	0,739271
Standardabweichung MSE	0,014330	0,029722	0,014816	0,005475	0,007392
Vergleich MSE	100,00%	89,08%	99,89%	100,87%	100,67%
Maximum MSE	0,748795	0,687955	0,750852	0,748684	0,747521
Minimum MSE	0,712746	0,597048	0,702363	0,734187	0,729671
Anzahl Simulationen	2020,00	884,00	1990,91	2020,00	2020,00
Vergleich Sim-anzahl	100,00%	43,76%	98,56%	100,00%	100,00%

Tabelle 4.15: GA - Verhältnis zwischen Crossover und Mutation - Modell 1

Aufgrund der teilweise sehr langen Laufzeiten der Optimierungen von Modell 2 muss auf eine Analyse des Crossover-Mutations-Verhältnisses für Modell 2 verzichtet werden.

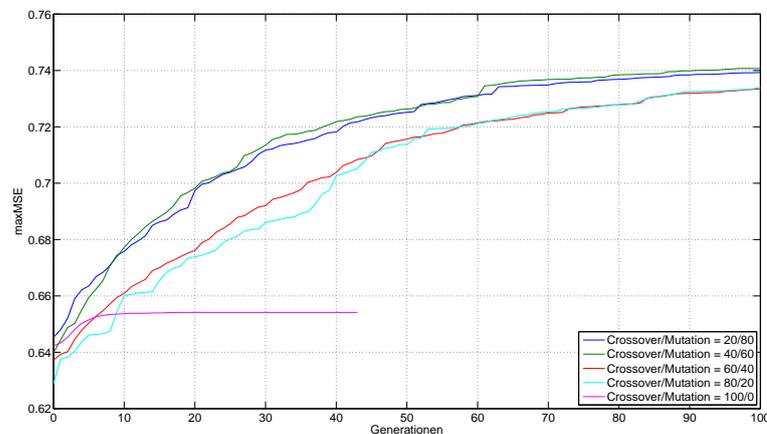


Abbildung 4.10: GA - Konvergenzverläufe unterschiedlicher Verhältniswerte zwischen Crossover und Mutation - Modell 1

#### 4.3.6.2 Ergebnis

Die Variation des Verhältnis zwischen dem Crossoveranteil und dem Mutationsanteil der Population zeigte für die Ergebnisse kaum nennenswerte Unterschiede. Lediglich die gemittelten Konvergenzverläufe der Optimierungen weisen geringfügige Differenzen auf. Die Optimierungen, in denen der Mutationsanteil deutlich größer als der Crossoveranteil ist, liefern unerwartet geringfügig bessere Ergebnisse. Das führt zu folgenden zwei Vermutungen:

1. Die genetische Vielfalt innerhalb der Population ist von Anfang an zu gering. Der Algorithmus wird erst durch die zusätzlichen genetischen Informationen, die durch die Mutation in die Population eingebracht werden, in die Lage versetzt, größere

Verbesserungen aufzuspüren. Das bedeutet, die zusätzlich generierten Informationen führen zu einer besseren Suchraumabdeckung.

2. Eine stärker zufällige Variation der Parameterwerte (wie sie auch beim Simulated Annealing eingesetzt wird) ist im vorliegenden Anwendungsfall die bessere Wahl als die strukturierte Variation der Werte nach vordefinierten Berechnungsgrundlagen.

Die erste Vermutung kann an dieser Stelle nicht weiter untersucht werden. Dazu wären weitere, sehr umfangreiche Tests von vor allem deutlich komplexeren Modellen nötig. Eine Überprüfung der zweiten Vermutung erfolgt in Abschnitt 4.5 beim Vergleich der Ergebnisse, die nach der Anpassung der implementierten Verfahren mittels der einzelnen Algorithmen erreicht werden.

## 4.4 Anpassen des PSO-Algorithmus

Für die Implementierung und Einbindung eines PSO-Algorithmus als Optimierungsalgorithmus in die Analyseumgebung wird die PSOt<sup>2</sup> Toolbox von Brian Birge eingesetzt [Bri03].

In den folgenden Abschnitten werden die verschiedenen Ablaufparameter des PSO-Algorithmus an das vorliegenden Optimierungsproblem angepasst. Das Ziel ist es, die bestmöglichen Ergebnisse bei einer möglichst geringen Zahl durchzuführender Simulationen zu erreichen. Wie bereits in Abschnitt 3.7 dargestellt gibt es eine Reihe von unterschiedlichen Parametern innerhalb des Algorithmus, die die Ergebnishüte und den Verlauf der Optimierungen beeinflussen können. Daher werden in diesem Kapitel umfangreiche Tests für die einzelnen Parameter durchgeführt, um eine möglichst stabile und gute Performance des Algorithmus zu gewährleisten.

Für die Tests werden alle Parameter auf einen zunächst festen Wert eingestellt. Zu Beginn der Tests werden die Werte der Parameter weitestgehend auf die empfohlenen Standardwerte der PSOt Toolbox gesetzt. Dann werden für jeden Parameter zahlreiche Analysen durchgeführt, in denen ein möglichst optimaler Parameterwert bestimmt wird.

Eine Übersicht über die Werte der einzelnen Parameter vor Beginn der Tests liefert die Tabelle 4.16.

---

<sup>2</sup>verfügbar unter <http://www.mathworks.com/matlabcentral/fileexchange/7506>

<b>Population Size</b>	20
<b>Max Iterations</b>	100
<b>ac1 - local best influence</b>	2
<b>ac2 - global best influence</b>	2
<b>inital inertia weight</b>	0,9
<b>final intertia weight</b>	0,4
<b>iteration, final inertia weight reached</b>	80
<b>min global error gradient</b>	0,00E+00
<b>epoch before error gradient terminates run</b>	30
<b>error goal</b>	NaN
<b>Type Flag</b>	0
<b>PSO seed</b>	0

**Tabelle 4.16:** Initiale Werte der Ablaufparameter des PSO

#### 4.4.1 Abbruchkriterien

Die Abbruchkriterien für die Suche nach der Parameterkonfiguration, die den größten Einfluss auf die Signalintegrität der zu analysierenden Bus-Modelle hat, werden analog der spezifizierten Abbruchkriterien des genetischen Algorithmus (siehe Abschnitt 4.3.1) definiert, um eine Vergleichbarkeit der Algorithmen zu ermöglichen. Das bedeutet, die Optimierungen terminieren jeweils nach

- 100 Iterationsrunden
- oder nach 30 Runden, wenn in diesen Runden keine Ergebnisverbesserung mehr gefunden werden konnte.

#### 4.4.2 Initialisierung der Partikel

Auch hier wird die Initialisierung der Partikel, wie die Initialisierung der Population des genetischen Algorithmus, dem Algorithmus selbst überlassen. Damit werden für beide Algorithmen die gleichen Ausgangsvoraussetzungen geschaffen, weil Vorkenntnisse oder Vermutungen über die Lage der gesuchten Optima nicht oder nur selten existieren und gerade die Eignung des Algorithmus für diese unbekanntenen Problemkonstellationen nachzuweisen ist.

#### 4.4.3 Populationsgröße

Ähnlich wie bei den genetischen Algorithmen bildet auch für die Partikel Schwarm Optimierung die richtige Wahl der Populationsgröße die Grundlage für eine erfolgreiche und performante Suche nach den globalen Optima eines Optimierungsproblems.

Aus diesem Grunde werden auch für die Partikel Schwarm Optimierung unterschiedliche Populationsgrößen getestet und ihr Einfluss auf die Qualität der gefundenen Ergebnisse untersucht und die benötigten Laufzeiten ausgewertet.

#### 4.4.3.1 Modell 1

In Analogie zu den Tests in Abschnitt 4.3.4.1 werden auch für die Populationsgröße der Partikel Schwarm Optimierung die Werte  $p = 5, 10, 15, \dots, 35$  getestet. Für jeden der Werte von  $p$  werden jeweils zehn Optimierungsläufe an Modell 1 durchgeführt. Eine Auflistung aller Einzelergebnisse findet sich im Anhang dieser Arbeit (siehe Tabelle C.1).

Die Tabelle 4.17 zeigt die über die jeweils zehn durchgeführten Analysen gemittelten Ergebnisse der Populations-Tests für Modell 1. Erwartungsgemäß werden die durchschnittlichen Ergebnisse der Analysen bei steigender Populationsgröße besser. Allerdings fallen die erreichten Verbesserungen mit einer Populationsgröße von  $p = 35$  im Vergleich zur Population von  $p = 5$  mit maximal 1,46 % recht gering aus. Die Zahl der jeweils durchgeführten Simulationen hingegen wächst enorm, um knapp 375 %, an.

Die minimale Verbesserung des Ergebnisses wird in diesem Falle also durch ein drastisches Anwachsen der Simulationszahlen und somit der Laufzeit der Analysen erkauft. Die Stabilität der erzielten Ergebnisse ändert sich ebenfalls mit wachsender Populationsgröße  $p$ . Die Standardabweichung im Fehlerwert sinkt, mit Ausnahme für  $p = 10$  kontinuierlich, und kann bei einer Populationsgröße von  $p = 35$  in etwa halbiert werden.

Die beschriebenen Ergebnisse führen zunächst zu der Vermutung, dass die Populationsgröße doch nicht einen so gravierenden Einfluss auf die Ergebnisse hat wie zunächst angenommen. Diese Vermutung wird vom Konvergenzverhalten der Optimierungen zusätzlich gestützt. Die Abbildung 4.11 zeigt den durchschnittlichen Konvergenzverlauf der Optimierungen für jede getestete Populationsgröße  $p$ . Deutliche Unterschiede in den Verläufen sind nur in den Kurven für die Populationsgrößen  $p = 5$  und  $p = 35$  zu erkennen. Während die Tests mit einer Populationsgröße von  $p = 5$  in den maxMSE-Werten lange Zeit gegenüber allen anderen deutlich zurückliegen, zeigen sich im Mittel über alle zehn durchgeführten Analysen für die größte getestete Populationsgröße  $p = 35$  die besten Ergebnisse im Verlauf der Optimierungen.

#### 4.4.3.2 Modell 2

Analog zu den an Modell 1 durchgeführten Analysen werden auch an Modell 2 identische Tests mit unterschiedlicher Populationsgröße  $p$  durchgeführt. Es werden wieder die Populationsgrößen  $p = 5, 10, 15, \dots, 35$  mit jeweils zehn durchgeführten Analysen getestet. Eine Auflistung aller Einzelergebnisse dieser Analysen liefert Tabelle C.2 im Anhang dieser Arbeit.

Bei Betrachtung der Ergebnisse zeichnet sich ein ähnliches Bild wie in den Optimierungsläufen an Modell 1 ab. Bis zu einer Populationsgröße von  $p = 20$  steigen die Ergebnisse im Schnitt um knapp 5 % an. Ab diesem Wert jedoch stagniert die weitere Verbesserung der gefundenen Ergebnisse. Die Schwankungen der Einzelergebnisse nimmt hingegen bei weiter ansteigenden Populationsgrößen weiter ab. Bei den Optimierungsläufen mit einer Popula-

tionsgröße von  $p = 35$  Individuen steigt die Standardabweichung noch einmal, entgegen den Erwartungen, leicht an und liegt in etwa wieder auf dem Niveau der Lösungen, die mit einer Populationsgröße von  $p = 20$  Partikeln erreicht wurden. Die gemittelten Ergebnisse, die über die zehn pro Populationsgröße  $p$  durchgeführten Analysen, werden in Tabelle 4.18 dargestellt.

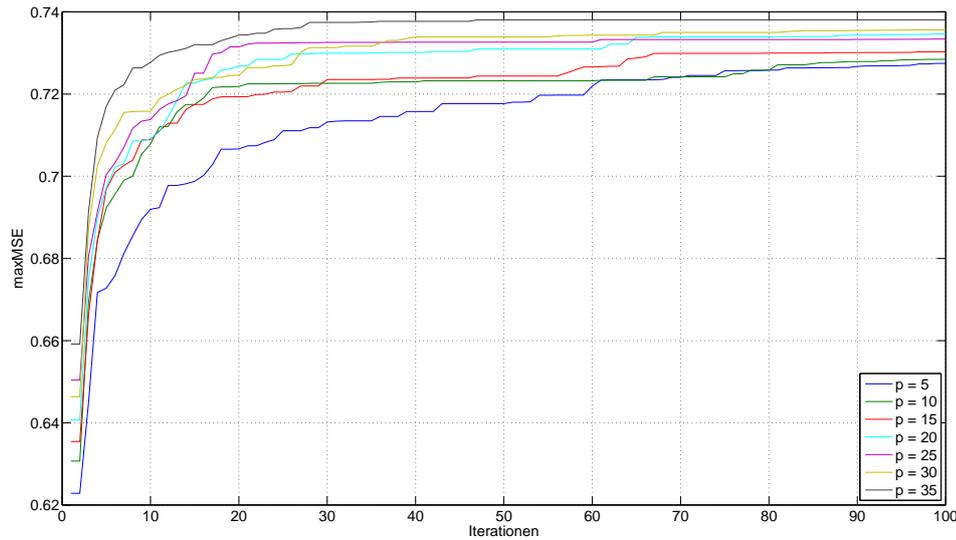


Abbildung 4.11: PSO - Populationsgröße: Konvergenzverhalten - Modell 1

Populationsgröße	5	10	15	20
Mittelwert maxMSE	0,7275	0,728504	0,7303	0,7346
Standardabweichung maxMSE	0,016390	0,012734	0,013898	0,011609
Vergleich Ergebnisse	100,00 %	100,14 %	100,39 %	100,98 %
Maximum maxMSE	0,7507	0,744378	0,7495	0,7505
Minimum maxMSE	0,6945	0,708669	0,7096	0,7165
c Anzahl Simulationen	383,75	508,00	906,92	1232,00
Vergleich Sim-Anzahl	100,00 %	132,38 %	236,33 %	321,04 %

Populationsgröße	25	30	35
Mittelwert maxMSE	0,733409	0,7357	0,7381
Standardabweichung maxMSE	0,006053	0,008075	0,007217
Vergleich Ergebnisse	100,82 %	101,13 %	101,46 %
Maximum maxMSE	0,743325	0,7473	0,7472
Minimum maxMSE	0,724813	0,7211	0,7227
∅ Anzahl Simulationen	1481,67	1825,00	1830,00
Vergleich Sim-Anzahl	386,10 %	475,57 %	476,87 %

Tabelle 4.17: PSO - Populationsgröße - Modell 1

Im Vergleich zu Modell 1 zeigen die durchschnittlichen Konvergenzverläufe der Optimierungsläufe ein etwas differenzierteres Verhalten (siehe Abbildung 4.12). Es fällt auf, dass vor allem die Ergebnisse, die mit den Populationsgrößen  $p = 5, 10, 15$  erreicht wurden, in ihrer Güte gegenüber den restlichen Analysen etwas abfallen. Die Ergebnisse für die großen Populationsgrößen ( $p = 25, 30, 35$ ) liegen fast über den gesamten Verlauf auf einem recht einheitlichen Niveau.

Populationsgröße	5	10	15	20
Mittelwert maxMSE	0,146667	0,150013	0,149121	0,153682
Standardabweichung maxMSE	0,005372	0,004326	0,004912	0,003441
Vergleich Ergebnisse	100,00 %	102,28 %	101,67 %	104,78 %
Maximum maxMSE	0,154242	0,154218	0,155765	0,156603
Minimum maxMSE	0,140032	0,143906	0,143437	0,145539
∅ Anzahl Simulationen	302,00	513,00	828,75	1174,55
Vergleich Sim-Anzahl	100,0 %	169,9 %	274,4 %	388,9 %

Populationsgröße	25	30	35
Mittelwert maxMSE	0,153115	0,153504	0,153354
Standardabweichung maxMSE	0,003002	0,002067	0,003578
Vergleich Ergebnisse	104,40 %	104,66 %	104,56 %
Maximum maxMSE	0,156987	0,157046	0,156903
Minimum maxMSE	0,145934	0,151658	0,145221
∅ Anzahl Simulationen	1438,64	1772,73	2089,50
Vergleich Sim-Anzahl	476,4 %	587,0 %	691,9 %

Tabelle 4.18: PSO - Populationsgröße - Modell 2

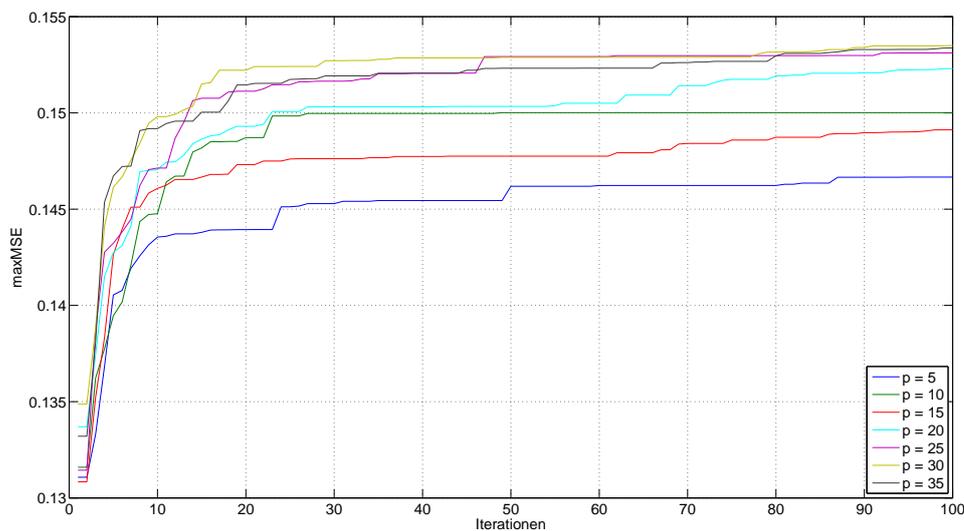


Abbildung 4.12: PSO - Populationsgröße: Konvergenzverhalten - Modell 2

### 4.4.3.3 Ergebnis

Die Optimierungsläufe mit unterschiedlichen Populationsgrößen zeigen, dass die Ergebnisse grundsätzlich bei größeren Populationen besser werden und bestätigen die anfänglichen Erwartungen. Allerdings ist der Zuwachs, der mit steigenden Populationsgrößen in der Ergebnisgüte zu erreichen ist, ab einer Populationsgröße von  $p = 20$  stark rückläufig. Durch die aber pro Runde zusätzlichen durchzuführenden Simulationen steigt ihre Gesamtzahl während der Optimierung und gleichzeitig die Gesamtlaufzeit weiter an.

Die Ergebnisse die an Modell 2 erzeugt wurden, zeigen, dass ab  $p = 20$  nahezu keine Ergebnisverbesserung mehr erreicht werden kann. Auch die Ergebnisse aus Modell 1 werden bei größeren, eingesetzten Populationen kaum noch besser. Die Wahl einer Populationsgröße von  $p = 20$  ist deshalb ein guter Kompromiss zwischen zu erreichender Ergebnisgüte und notwendiger Zahl an Simulationsläufen. Für die weiteren Analysen und Schritte der Algorithmusanpassung ist daher die ausschließliche Verwendung einer Populationsgröße von  $p = 20$  gerechtfertigt.

## 4.4.4 Trägheitsgewichtung

Bei der Adaption eines PSO-Algorithmus an unterschiedliche praktische Optimierungsprobleme stellt die Konfiguration der Parameter für die Trägheitsgewichtung (siehe Abschnitt 3.7.4 und 3.7.5) der Partikel einen entscheidenden Aspekt dar. Shi und Eberhart konstatieren in [SE98a] der Trägheitsgewichtung eine ausschlaggebende Rolle für die Konvergenz der Partikel Schwarm Optimierung.

In den vielen Fällen wird die Trägheitsgewichtung der Partikel in Form eines im Laufe der Optimierung langsam sinkenden Parameters realisiert [OBP02, SE98b]. Diese Form der Implementierung hat sich vielfach als effizienter erwiesen als eine Festwertparametrisierung. Daher wird auch hier eine lineare Dekrementierung der Trägheitsgewichtung bis zu einem gegebenen finalen Wert eingesetzt. Im Folgenden sollen nun die entsprechenden Parameter bestmöglich adaptiert werden.

Aus den Optimierungsläufen an den Testmodellen in Abschnitt 4.4.3 und den in Tabelle 4.16 angegebenen Algorithmusparametern ist ersichtlich, dass die Suche nach der Parameterkonfiguration der Bussysteme in nahezu allen Fällen für die gewählten Populationsgröße  $p = 20$  terminiert, bevor der finale Wert der Trägheitsgewichtes erreicht wird. Die Bewegungen der Partikel sind demnach weniger stark von der im Schwarm gefundenen besten Lösung abhängig als dies durch die finale Trägheitsgewichtung vorgesehen ist. Das bedeutet, die Neuberechnungen der Partikelbewegungen sind stark abhängig von der bisherigen Bewegung der Partikel und führen somit zu einem deutlich langsameren Konvergieren des Schwarms.

Im Folgenden werden für die beiden Test-Modelle Analysen durchgeführt, bei denen der finale Trägheitswert in einer früheren Iterationsrunde erreicht wird. Auf diese Weise soll

ein früheres Konvergieren des Schwarms und somit eine bessere lokale Exploration der Umgebung, der jeweils aktuell gefundenen besten Lösung, erreicht werden. Gleichzeitig muss jedoch ein zu frühes Konvergieren vermieden werden, damit der Algorithmus nicht vorzeitig in einem lokalen Optimum terminiert.

#### 4.4.4.1 Modell 1

Um den Einfluss der Trägheitsgewichtung zu bestimmen, werden zunächst der initiale Wert  $iw = 0,9$  des Parameters sowie der finale Wert  $fiw = 0,4$  beibehalten und nur die Iterationsstufe, in der der finale Wert erreicht wird, schrittweise verringert. Konkret wird ein Erreichen der finalen Gewichtung in den Iterationsrunden  $n = 10, 20, \dots, 70$  getestet. Für jeden Wert werden zunächst fünf Analysen durchgeführt.

Es zeigen sich erkennbare Unterschiede bei den Mittelwerten der erreichten Ergebnisse. Die Analysen, in denen der finale Werte des Trägheitsparameters früher erreicht wird ( $n = 10, 20, 30, 40$ ), liefern bessere Ergebnisse als die übrigen Analysen. Da die Ergebniswerte für die vier genannten Parameterwerte alle recht nahe beieinander liegen, werden für alle vier Werte zusätzlich fünf weitere Analyseläufe durchgeführt, um die Genauigkeit der gemittelten Ergebnisse zu erhöhen. Die Ergebnisse, die die Analysen im Mittel über alle zehn Durchläufe erreicht haben, werden in Tabelle 4.19 dargestellt. Als Referenz zum Vergleich der Ergebnisse untereinander dienen die Ergebnisse aus Abschnitt 4.4.3.1, die mit entsprechender Populationsgröße und dem Erreichen der finalen Trägheitsgewichtung in Iteration  $n = 80$  erzielt wurden.

Deutlich fallen auch die Unterschiede im Konvergenzverhalten aus, wie die Abbildung 4.13 veranschaulicht. Um den Nutzen zu beurteilen, den das frühere Erreichen des finalen Trägheitswertes erzielt, ist der Konvergenzverlauf für den Wert  $n = 80$  zusätzlich mit in die Abbildung aufgenommen worden. Es ist zu sehen, dass nach anfänglich vergleichbarer Konvergenz der Analysen die Ergebnisse der Analysen mit  $n = 80$  ab etwa Iteration 25 nur noch sehr langsam steigen. Dies ist der Zeitpunkt zu dem scheinbar eine gewisse Ergebnisgrenze gefunden wurde bis zu der viele Parameterkonfigurationen existieren, die weit über den Suchraum verstreut sind. Eine weitere Verbesserung ist offensichtlich deutlich schwerer zu finden. An dieser Stelle profitieren die Analysen mit kleineren Werten für  $n$  von der stärkeren Beeinflussung durch die global gefundenen Lösungen für die Neuberechnung der Partikelbewegungen. Während in den Analysen mit  $n = 80$  die Partikel noch vergleichsweise stark ihre alten Bewegungsrichtungen beibehalten, werden die Partikel in den Analysen mit niedrigeren Werten für  $n$  offensichtlich stärker von den im Schwarm gefunden Lösungen in ihren Bewegungen beeinflusst. Somit werden im letzten Fall die Umgebungen der guten Lösungen genauer und besser nach weiteren kleineren Verbesserungen abgesucht.

Eine weniger stark ausgeprägte Individualität der Bewegungen der Partikel führt offensichtlich zu verbesserten Ergebnissen bei der Optimierung.

Aufgrund der beschriebenen Ergebnisse werden zusätzlich noch jeweils zehn Analysen durchgeführt, die nach  $n = 10$  bzw.  $n = 20$  einen finalen Wert  $fiw = 0,2$  erreichen. Die Ergebnisse finden sich in Tabelle 4.20. In den Mittelwerten der erzielten Ergebnisse unterscheiden sich die Analysen nur wenig, weiterhin benötigen alle Analysen dieselbe Zahl an Simulationen. Ein Unterschied zeigt sich lediglich in der Stabilität der Mittelwerte. So ist die Abweichung der einzelnen Ergebnisse vom Mittelwert bei Verwendung des reduziertes finalen Trägheitswerts  $fiw = 0,2$  nur etwa halb so groß. Die einzelnen Ergebnisse aller hier durchgeführten Optimierungsläufe finden sich im Anhang dieser Arbeit (siehe Tabelle C.3).

n	80	10	20	30	40
Mittelwert maxMSE	0,734612	0,745298	0,747727	0,744476	0,743835
Standardabweichung maxMSE	0,011609	0,001588	0,003216	0,003732	0,009363
Vergleich maxMSE	100,00 %	101,45 %	101,79 %	101,34 %	101,26 %
Maximum maxMSE	0,750450	0,747516	0,752861	0,749915	0,750651
Minimum maxMSE	0,716457	0,743960	0,742711	0,737702	0,724235
∅ Anzahl Simulationen	1232	2020	2020	1796	1786
Vergleich Sim-anzahl	100,00 %	163,96 %	163,96 %	145,78 %	144,97 %

Tabelle 4.19: PSO - Trägheitsgewicht - Modell 1

fiw	0,4	0,2	
n	20	10	20
Mittelwert maxMSE	0,747727	0,748322	0,748048
Standardabweichung maxMSE	0,003216	0,002259	0,001524
Vergleich maxMSE	100,00 %	100,08 %	100,04 %
Maximum maxMSE	0,752861	0,751186	0,750501
Minimum maxMSE	0,742711	0,746080	0,745950
∅ Anzahl Simulationen	2020	2020,00	2020,00
Vergleich Sim-Anzahl	100,00 %	100,00 %	100,00 %

Tabelle 4.20: PSO - Trägheitsgewicht:  $fiw = 0,2$  - Modell 1

#### 4.4.4.2 Modell 2

Zur Überprüfung der Erkenntnisse aus Modell 1, werden auch an Modell 2 die Werte  $n = 10, 20, 30, 40$  für die Iterationsrunde getestet, in denen der finale Trägheitswert erreicht wird. Außerdem wird zusätzlich noch die Wertekombination  $n = 20, fiw = 0,2$  überprüft. Für alle Werte werden jeweils fünf Analysen durchgeführt. Eine Auflistung der Ergebnisse aller einzelnen Optimierungsläufe findet sich im Anhang in Tabelle C.4. Die gemittelten Ergebnisse sind in Tabelle 4.21 zu finden. Auch hier werden die besten Ergebnisse von den Analysen erreicht, in denen der finale Trägheitswert früher erreicht wird. Für  $n = 10$

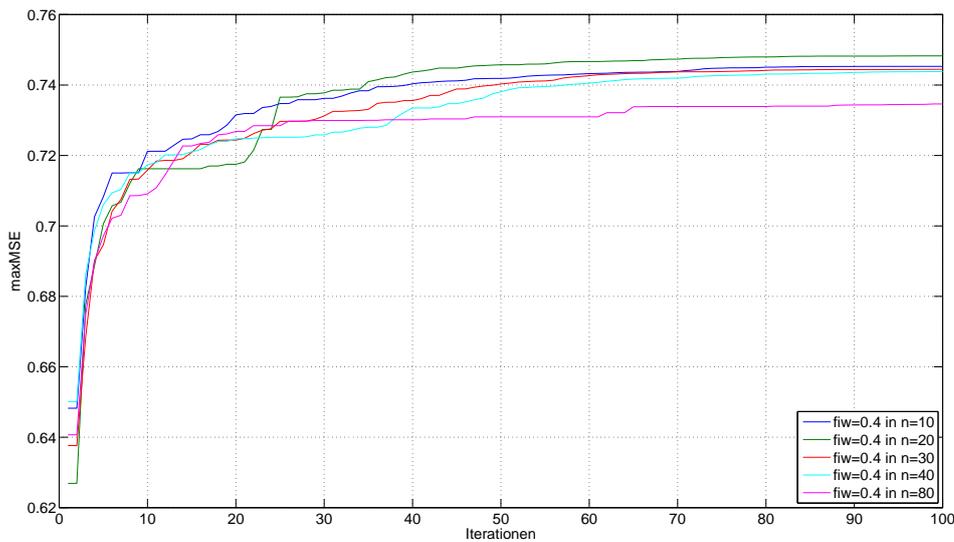


Abbildung 4.13: PSO - Trägheitsgewicht - Modell 1

und  $n = 20$  lässt sich eine Verbesserung der Ergebnisse um etwa 2 % erreichen. Die Zahl der durchgeführten Simulationen wächst dabei allerdings um mindestens knapp 50 % an. Ebenfalls wie in den Analysen, die an Modell 1 durchgeführt worden sind, reduziert sich bei Verwendung von  $fiw = 0,2$  die Varianz der Ergebnisse. Der Zuwachs an Simulationen hingegen ist im Vergleich zu den anderen Analysen immens. In den Konvergenzverläufen zu den Analysen (siehe Abbildung 4.14) sind die Zeitpunkte, an denen der finale Trägheitswert erreicht wird, gut zu erkennen. Im Verlauf aller Analysen ist kurz vor Erreichen des jeweiligen finalen Trägheitswertes ein deutlicher Anstieg der Kurvenverläufe erkennbar.

n	80	10	20	30	40	20, $fiw = 0,2$
Mittelwert maxMSE	0,153682	0,156883	0,156743	0,156268	0,154793	0,156914
Standardabweichung maxMSE	0,003441	0,001135	0,001259	0,000626	0,001873	0,000848
Vergleich maxMSE	100,00 %	102,08 %	101,99 %	101,68 %	100,72 %	102,10 %
Maximum maxMSE	0,156603	0,158068	0,157457	0,157025	0,157001	0,157804
Minimum maxMSE	0,145539	0,155279	0,154501	0,155426	0,152964	0,155999
∅ Anzahl Simulationen	1174,55	2020,00	1752,00	1904,00	1304,00	2020,00
Vergleich Sim-Anzahl	100,00 %	171,98 %	149,16 %	162,11 %	111,02 %	171,98 %

Tabelle 4.21: PSO - Trägheitsgewicht - Modell 2

#### 4.4.4.3 Ergebnisse

Die durchgeführten Analysen haben gezeigt, dass ein früheres Erreichen des finalen Trägheitswertes zu einer Steigerung des gefundenen maxMSE-Werte um fast 2 % führt. Dazu werden allerdings etwa gut 60 % mehr Simulationen durchgeführt. Weiterhin ist bei einer

Reduzierung des finalen Trägheitswertes von  $fiw = 0,4$  auf  $fiw = 0,2$  eine größere Stabilität der Ergebnisse zu erreichen. Für die weitere Anpassung des Algorithmus wird folgende Kombination der Parameter zur Steuerung des Trägheitsgewichts eingesetzt:  $n = 20$ ,  $iiw = 0,9$ ,  $fiw = 0,2$ .

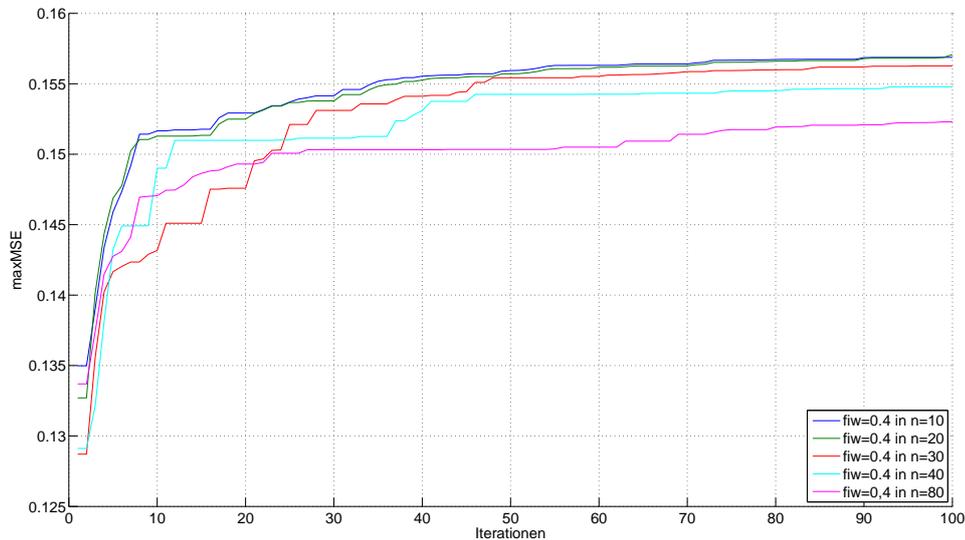


Abbildung 4.14: PSO - Trägheitsgewicht - Modell 2

## 4.4.5 Beschleunigungskoeffizienten

Nachdem die Parameter der Trägheitsgewichtung angepasst wurden, bieten die beiden Beschleunigungskoeffizienten die nächste Möglichkeit direkten Einfluss auf die Partikelbewegungen zu nehmen (siehe Abschnitt 3.7.5).

### 4.4.5.1 Modell 1

Es werden wieder mehrfache Testläufe mit unterschiedlichen Wertekonfigurationen für die beiden Parameter durchgeführt. Die resultierenden Ergebnisse werden in der Tabelle 4.22 dargestellt. Es ist zu beobachten, dass keine der Wertekombinationen wesentliche Unterschiede zu den Default-Werten  $ac1 = 2$  und  $ac2 = 2$  zeigt. Ebenso zeigen sich nur minimale Unterschiede in der Anzahl der jeweils durchgeführten Simulationen, insbesondere ergeben sich Einsparungen nur, wenn die erzielten Ergebnisse ebenfalls geringer als der Referenzwert ausfallen. Die auftretenden Unterschiede können daher vernachlässigt werden.

Die Auflistung der Ergebnisse der einzelnen Analyseläufe findet sich in Tabelle C.5 in Anhang.

ac1	2					
ac2	2	1,8	2,2	2,4	3	1
Mittelwert maxMSE	0,748048	0,747342	0,748180	0,748335	0,747025	0,736706
Standardabweichung maxMSE	0,001524	0,001131	0,002803	0,002616	0,002320	0,002522
Vergleich maxMSE	100,00 %	99,91 %	100,02 %	100,04 %	99,86 %	98,48 %
Maximum maxMSE	0,750501	0,748494	0,752290	0,753270	0,749892	0,739618
Minimum maxMSE	0,745950	0,744883	0,745000	0,744260	0,744381	0,735216
∅ Anzahl Simulationen	2020,00	1978,00	1960,00	1971,67	1973,33	1746,67
Vergleich Sim-Anzahl	100,00 %	97,92 %	97,03 %	97,61 %	97,69 %	86,47 %

ac1	1,8	2,2
ac2	2	
Mittelwert maxMSE	0,746949	0,748181
Standardabweichung maxMSE	0,001446	0,003115
Vergleich maxMSE	99,85 %	100,02 %
Maximum maxMSE	0,748805	0,751663
Minimum maxMSE	0,744749	0,744900
∅ Anzahl Simulationen	2020,00	2020,00
Vergleich Sim-Anzahl	100,00 %	100,00 %

Tabelle 4.22: PSO - Beschleunigungskoeffizienten - Modell 1

#### 4.4.5.2 Ergebnis

Da eine Veränderung der Beschleunigungskoeffizienten offensichtlich nur sehr geringen Einfluss auf die Ergebnisse und die Laufzeiten der Bussystem-Analyse hat, wird auf eine weitere Untersuchung ihres Einflusses verzichtet. Stattdessen werden im weiteren die Standardwerte für beide Parameter ( $ac1 = 2$  und  $ac2 = 2$ ) beibehalten.

#### 4.4.6 Umgang mit Wertebereichsgrenzen

In Abschnitt 3.7.5 wurden zwei unterschiedliche Ansätze vorgestellt, mit denen eine Berücksichtigung von Wertebereichsgrenzen einzelner Parameter innerhalb einer Optimierung mittels des PSO-Algorithmus realisiert werden kann. Zum einen besteht die Möglichkeit die Partikel, wenn sie auf ihrem Flug durch den Suchraum eine solche Grenze erreichen, an dieser abprallen zu lassen und sie zurück in den Suchraum zu reflektieren. Diese als *bounce-off* bezeichnet Methode fand in den bisherigen Analysen des PSO Verwendung. An dieser Stelle wird nun der Vergleich zur zweiten möglichen Methode gezogen. Sie besteht darin für den Parameter, dessen Wertebereichsgrenze erreicht wurde, in der jeweiligen Iterationsrunde den Wert des Parameters auf den erreichten Grenzwert zu fixieren. Diese Umsetzung wird als *saturation-at-limit* bezeichnet.

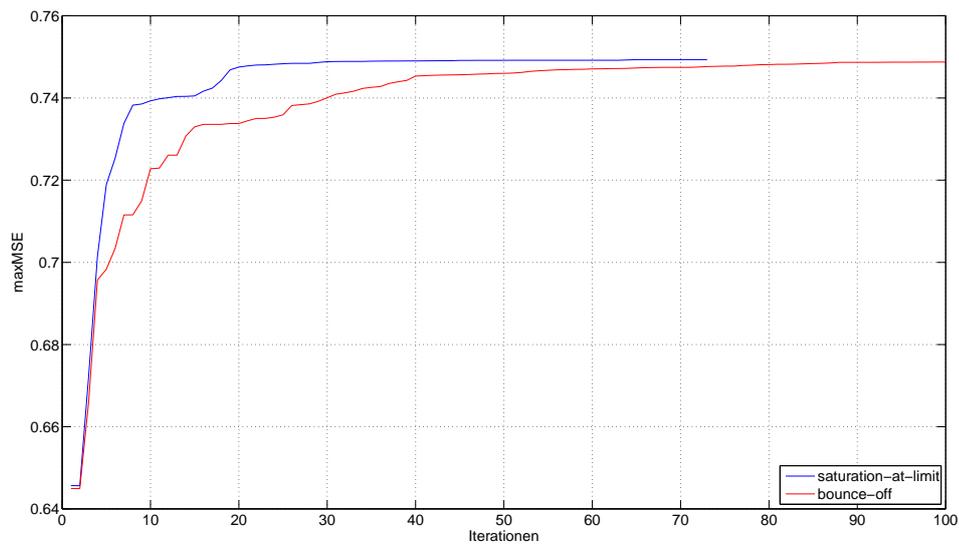
##### 4.4.6.1 Modell 1

Für jede der beiden Möglichkeiten werden jeweils zehn Analyseläufe durchgeführt. Die Ergebnisse sind in der Tabelle 4.23 dargestellt. Hier zeigen sich deutlichere Auswirkungen auf

Methode	bounce-off	saturation at limit
Mittelwert maxMSE	0,748048	0,749334
Standardabweichung maxMSE	0,001524	0,002239
Vergleich maxMSE	100,00 %	100,17 %
Maximum maxMSE	0,750501	0,751983
Minimum maxMSE	0,745950	0,745825
AØ Anzahl Simulationen	2020	1486
Vergleich Sim-Anzahl	100,00 %	73,56 %

**Tabelle 4.23:** PSO - Umgang mit Wertebereichsgrenzen - Modell 1

die Ergebnisse der Analysen. Die Methode, die den Parameter auf seiner Wertebereichsgrenze fixiert, erzielt ebenso gute Ergebnisse wie die bounce-off-Methode, allerdings wird die Zahl der Simulationen, die benötigt wird, um diese Lösungen zu finden, um knapp 30 % reduziert, nämlich von durchschnittlich 2020 auf 1466 Simulationen. Eine schnellere, aber im Ergebnis gleichwertige Konvergenz ist auch in der graphischen Darstellung der Konvergenzverläufe beider Verfahren in Abbildung 4.15 deutlich erkennbar.



**Abbildung 4.15:** PSO - Umgang mit Parameterwertebereichsgrenzen: Konvergenzverhalten - Modell 1

Die saturation-at-limit Methode, bei der die Parameterwerte auf die Wertebereichsgrenzen fixiert werden, findet sehr viel schneller gute Lösungen als die bounce-off Methode. Bereits nach 20 Iterationsrunden liegt die Güte der gefundenen Lösungen auf dem Niveau, das bei Verwendung der bounce-off-Methode erst am Ende der Optimierungen erreicht wird. Bis etwa zur 30. Iterationsrunde lassen sich noch weitere kleinere Verbesserungen finden.

Methode	bounce-off	saturation-at-limit
Mittelwert maxMSE	0,156914	0,157383
Standardabweichung maxMSE	0,000848	0,001319
Vergleich maxMSE	100,00 %	100,30 %
Maximum maxMSE	0,157804	0,159005
Minimum maxSE	0,155999	0,155239
∅ Anzahl Simulationen	2020,00	1863,64
Vergleich Sim-Anzahl	100,00 %	92,26 %

**Tabelle 4.24:** PSO - Umgang mit Wertebereichsgrenzen - Modell2

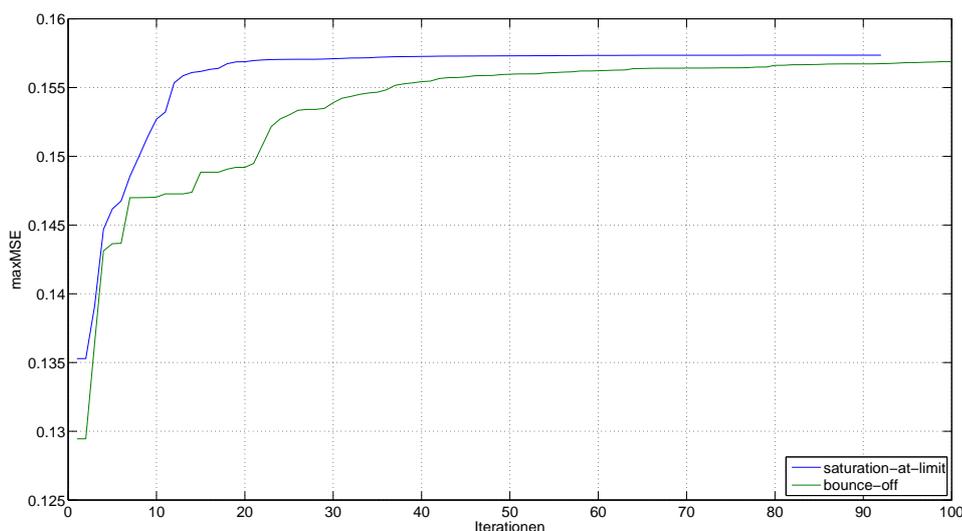
Danach läuft die Suche mit Hilfe der saturation-at-limit Methode 2 im Schnitt zwar noch bis zur 73. Iteration, die noch erzielte Steigerung des maxMSE-Wertes ist aber nur sehr gering.

Eine Auflistung der Ergebnisse aller Analyseläufe findet sich im Anhang in Tabelle C.6.

#### 4.4.6.2 Modell 2

Es wurden auch an Modell 2 jeweils zehn Analysen mit den beiden Verfahren zum Umgang mit Wertebereichsgrenzen durchgeführt. Die detaillierten Ergebnisse der einzelnen Analysen finden sich im Anhang (siehe Tabelle C.7). Die zu beobachtenden Ergebnisse zeigen ein ähnliches Bild wie die Analysen des ersten Modells. Allerdings unterscheiden sich die Ergebnisse der Analysen von Modell 2 weniger stark. Die Ergebnisse, die beide Verfahren im Mittel für die MSE-Fehlerwerte erreichen, liegen auf identischem Niveau (siehe Tabelle 4.24). Der Vergleich der Zahl durchgeführter Simulationen fällt wieder zu Ungunsten der bounce-off-Methode aus, allerdings im Hinblick auf die 30 % Einsparung bei Modell 1, mit lediglich gut 7,5 %.

Deutlicher unterscheiden sich hingegen wieder die Konvergenzverläufe der beiden Verfahren (siehe Abbildung 4.16). Wird der Parameter, dessen Wertebereichsgrenze erreicht wird, auf diesen Wert fixiert, sind gute Ergebnisse wesentlich schneller erreichbar, als wenn der Partikel an der Wertebereichsgrenze abprallt und in der Suchraum reflektiert wird. Bereits nach zwanzig Iterationsrunden hat die gefundene Lösung einen Wert von  $maxMSE = 0,157$  erreicht. Im restlichen Verlauf der Optimierung werden dann nur noch sehr geringe Verbesserungen erzielt. Die bounce-off-Methode hat nach zwanzig Iterationen gerade ein mal einen Ergebniswert von  $maxMSE = 0,1492$  erreicht. Diese Ergebnisdifferenz zwischen beiden Verfahren verringert sich im Fortlauf der Optimierung zwar zunehmend, kann jedoch bis Erreichen der hundertsten Iteration nicht komplett ausgeglichen werden. Somit liefert die saturation-at-limit-Methode nicht nur viel schneller gute Ergebnisse, sondern am Ende auch das bessere Endresultat.



**Abbildung 4.16:** PSO - Umgang mit Parameterwertebereichsgrenzen: Konvergenzverhalten - Modell 2

#### 4.4.6.3 Ergebnis

Es ist deutlich an den erzielten Ergebnissen zu erkennen, dass der Ansatz bei dem die Partikel in den Suchraum reflektiert werden, die schlechtere Auswahl im vorliegenden Anwendungsfall ist.

Dieses Verhalten lässt sich damit begründen, dass die Wertekombinationen der Busparameter, die zu einer stärkeren Beeinträchtigung der Signale auf dem Bus und damit zu einer höheren MSE-Bewertung führen, eher in den Randbereichen der Werteintervalle der Variationsparameter des Busmodells zu finden sind. Eine Fixierung der Parameter auf ihrer Wertebereichsgrenze führt sehr viel schneller zum Auffinden von Parameterkonfigurationen, deren zugehörige maxMSE-Werte recht nahe an den maximal gefundenen Lösungen liegen.

## 4.5 Vergleich der Ergebnisse

Im folgenden Abschnitt werden nun die erzielten Optimierungen der drei Algorithmen miteinander verglichen und die Ergebnisse bewertet.

### 4.5.1 Modell 1

Für das kleinere der beiden Testmodelle ergibt sich aus dem Vergleich der Ergebnisse, die mit den drei Algorithmen erzielt wurden, folgender Sachverhalt:

Die Bussystemanalysen, die für die Parametervariation den Simulated Annealing Algorithmus eingesetzt haben, erzielen mit minimalen Vorsprung den besten maxMSE-Wert. Im

Detail wird eine Steigerung des maxMSE-Wertes im Mittel um 146,48 % gegenüber dem Eingangswert in Abschnitt 4.1.1 erreicht. Durch den Einsatz der Lokalisierungsschritte wird zusätzlich die Laufzeit der Optimierungen, durch Reduzierung der Zahl durchgeführter Simulationen um etwa 60 %, enorm verkürzt (siehe Abschnitt 4.2.6.1). Im Schnitt benötigt der Simulated Annealing Algorithmus nach der erfolgten Anpassung der Parameter 269 Simulationen für die Bestimmung seiner Lösung.

Weil der genetische Algorithmus und die Partikel Schwarm Optimierung auf Populationen basieren, sind die Laufzeiten dieser beiden Verfahren wesentlich höher. Der genetische Algorithmus benötigt im Schnitt 2020 Simulationen bis er terminiert. Die Partikel Schwarm Optimierung kommt hingegen mit einer etwas geringeren Zahl an Simulationen (1486 Simulationen) aus. Bei der gegebenen Komplexität von Modell 1 können beide Algorithmen das Potential ihres parallelen Suchansatzes nicht ausnutzen. Die ermittelten maxMSE-Werte beider Verfahren liegen jeweils nur knapp unter denen des Simulated Annealing, mit leichten Vorteilen für die Partikel-Schwarm-Optimierung.

Die Tabelle 4.25 liefert eine Gegenüberstellung der erreichten Ergebnisse und verdeutlicht das performantere Verhalten im Hinblick auf die Laufzeit der Optimierungen des Simulated Annealing noch einmal eindrucksvoll.

Algorithmus	SA	GA	PSO
<b>Referenz MSE ohne Optimierung</b>	0,514374		
<b>Mittelwert maxMSE</b>	0,753457	0,740720	0,749334
<b>Standardabweichung maxMSE</b>	0,002004	0,002004	0,002239
<b>Vergleich maxMSE</b>	146,48 %	144,00 %	145,68 %
<b>Maximum maxMSE</b>	0,754933	0,748684	0,751983
<b>Minimum maxMSE</b>	0,748203	0,734187	0,745825
<b>∅ Anzahl Simulationen</b>	269	2020	1486

**Tabelle 4.25:** Auswertung der Optimierungsergebnisse der Algorithmen - Modell 1

## 4.5.2 Modell 2

Bei der Auswertung der Ergebnisse für Modell 2 ergibt sich ein ähnliches Bild wie bei Modell 1.

Mit minimalem Vorsprung liegen die Ergebnisse aus den Analyseläufen, die unter Verwendung des Simulated Annealing durchgeführt werden, über den maxMSE-Werten, die mittels der Partikel Schwarm Optimierung erzielt werden (siehe Tabelle 4.26). Beide Verfahren erreichen eine Steigerung des MSE-Wertes um gut 40 % gegenüber dem eingangs in Abschnitt 4.1.2 bestimmten maxMSE-Wert des Modells. Die Ergebnisse aus den Analyseläufen, bei denen die Parametervariation mit dem genetischen Algorithmus gesteuert wurde, reichen, ähnlich wie bei Modell 1, nicht ganz die Ergebniswerte der übrigen beiden Verfahren heran. Allerdings fallen die Differenzen in den Ergebniswerten hier jedoch

deutlicher aus. Die Ergebnisse, die mittels des genetischen Algorithmus ermittelt wurden, erreichen lediglich eine Steigerung von gut 25 % gegenüber den Fehlerwerten aus der Analyse des Bussystem mit der initialen Parameterkonfiguration in Abschnitt 4.1.2.

In Bezug auf die Laufzeiten der Optimierungsläufe ergibt sich ein vergleichbares Bild

Algorithmus	SA	GA	PSO
<b>Referenz MSE ohne Optimierung</b>	0,112180		
<b>Mittelwert maxMSE</b>	0,158093	0,140931	0,157383
<b>Standardabweichung maxMSE</b>	0,001333	0,001371	0,001319
<b>Vergleich maxMSE</b>	140,93 %	125,63 %	140,29 %
<b>Maximum maxMSE</b>	0,159798	0,141852	0,159005
<b>Minimum maxMSE</b>	0,156425	0,139355	0,155239
<b>Anzahl Simulationen</b>	744	2020,00	1864

**Tabelle 4.26:** Auswertung der Optimierungsergebnisse der Algorithmen - Modell 2

zu Modell 1. Die wenigsten Simulationen während der Analysen, und somit die kürzeste Laufzeit, benötigt der Simulated Annealing Algorithmus. Er kommt trotz der deutlich gestiegenen Modellkomplexität durch Einsatz von Lokalisierungsschritten im Durchschnitt mit 744 Simulationen aus. Gegenüber dem kleinen Modell ist hier allerdings ein deutlicher Anstieg der Simulationsanzahl zu verzeichnen. Dies resultiert aus der Tatsache, dass die Anzahl der Folgelösungen, die berechnet werden, ehe der Kontrollparameter abgesenkt wird, direkt von der Anzahl der variablen Parameter im Busmodell abhängt. Die Laufzeit der Partikel Schwarm Optimierung erhöht sich ebenfalls, allerdings nicht ganz so stark wie beim Simulated Annealing. Zur Bestimmung der Kombination der variablen Parameter des Busmodells, die den größten maxMSE-Wert erzeugt, werden im Schnitt 1864 Simulationen benötigt. Tendenziell ist zu erkennen, dass sich die Laufzeiten bei steigender Modellkomplexität zwischen den populationsbasierten, parallelen Suchansätzen (GA und PSO) und dem sequentiellen Suchansatz des Simulated Annealing scheinbar angleichen. Es ist zu annehmen, dass sich die Laufzeiten bei weiter steigender Modellkomplexität weiter annähern werden. In weiteren Untersuchungen wäre zu überprüfen, ob sich das Laufzeitverhalten ab einem bestimmten Komplexitätsgrad der Modelle zugunsten der populationsbasierten Verfahren ändert.

### 4.5.3 Auswertung

Im Anschluss an die gezeigten Ergebnisse nach der Anpassung der Algorithmen stellt sich die Frage, welches der implementierten Verfahren am besten für den Einsatz zur Parametervariation innerhalb der Kfz-Bussystemanalyse geeignet ist. Die reinen Ergebniswerte, die sich aus den Optimierungsläufen mit den verschiedenen Verfahren ergeben haben, wurden in den Abschnitten 4.5.1 und 4.5.2 präsentiert. In den erzielten maxMSE-Werten unterscheiden sich die Analysen mittels Simulated Annealing und der Partikel Schwarm

Optimierung nur geringfügig. Die Laufzeiten der Verfahren hingegen weisen zum Teil sehr große Unterschiede auf. Bei Vergleich der jeweiligen Konvergenzverläufe relativiert sich dieses Bild allerdings wieder. Während sich bei der Betrachtung der Konvergenzverläufe für die Analysen mittels Simulated Annealing und Partikel Schwarm Optimierung für Modell 1 nur geringe Unterschiede, mit Vorteilen für die Konvergenzgeschwindigkeit des Simulated Annealing zeigen (siehe Abbildung 4.17), ändert sich dies bei steigender Modellkomplexität im Modell 2.

Abbildung 4.18 zeigt die durchschnittlichen Konvergenzverläufe der Bussystem-Analysen für Modell 2 nach der erfolgten Anpassung der Optimierungsverfahren. Es ist zu erkennen, dass die Analysen mittels der Partikel Schwarm Optimierung zu Beginn einen deutlich schnelleren Anstieg in den erreichten Ergebniswerten aufweisen. Erst nach etwa 400 durchgeführten Simulationen erreicht der Simulated Annealing Algorithmus annähernd das Ergebnisniveau, das die Partikel Schwarm Optimierung zeitgleich annimmt. Erst weitere 200 Simulationen später übersteigen die maxMSE-Werte aus den mittels Simulated Annealing durchgeführten Analysen die Ergebnisse der Partikel Schwarm Optimierung.

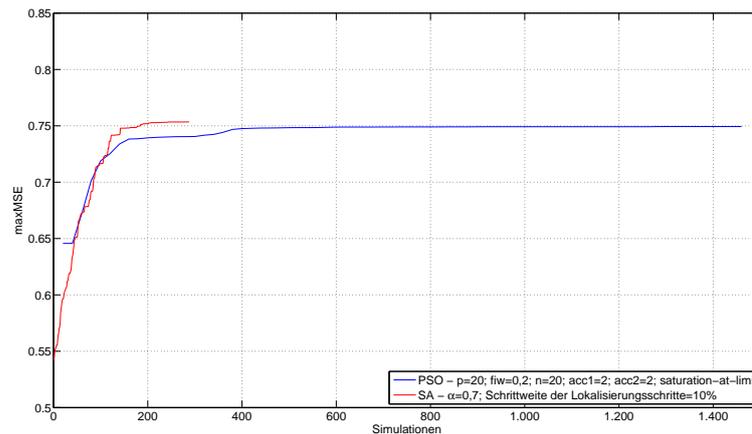
Es lassen sich folgende Schlüsse aus den beschriebenen Tatsachen ziehen: Bei kleinen Modellen ist der parallele Suchansatz der Partikel Schwarm Optimierung zu komplex. Für die einzelnen Verbesserungsschritte ist immer die komplette Population mit  $p$  Individuen zu bewerten, es können also nur jeweils alle  $p$  Simulationen Verbesserungen erzielt werden. In dieser Zeit hat der Simulated Annealing Algorithmus insgesamt  $p$ -mal die Möglichkeit, eine verbesserte Lösung zu bestimmen und von dieser Lösung aus weiter zu suchen. Bei deutlich komplexeren Modellen kehrt sich dieses Verhältnis zunehmend um. Durch den parallelen Suchansatz ist die Partikel Schwarm Optimierung in der Lage den Suchraum gleichzeitig von mehreren Stellen aus zu durchsuchen. Somit kann durch eine schnelle, großflächige Abtastung des Suchraumes die ungefähre Lage des zu findenden Optimums bestimmt werden. Der Simulated Annealing Algorithmus benötigt hingegen eine gewisse Zahl an Simulationen bis er sich schrittweise in die Nähe des gesuchten Optimums bewegt hat.

Je nach Modellkomplexität kann also der Einsatz des Simulated Annealing oder der Partikel Schwarm Optimierung angeraten sein, um schnell möglichst gute Ergebnisse zu bestimmen. Bei ausreichend langer Suchzeit führen beide Verfahren zu annähernd identischen Ergebnissen.

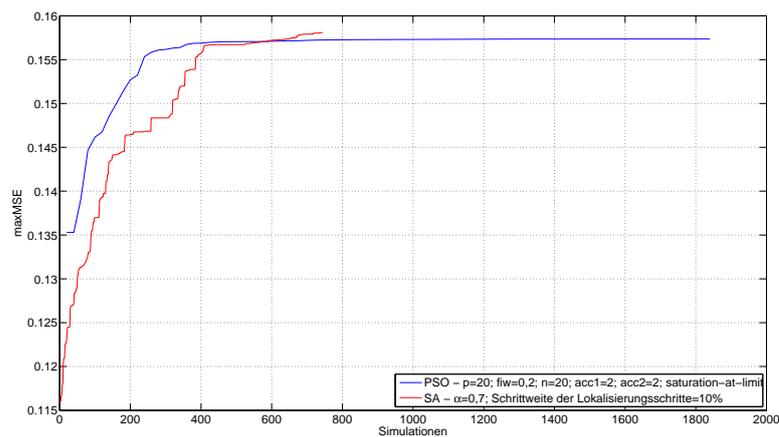
Das Verfahren der Partikel Schwarm Optimierung bietet sich vor allem für die Analyse hoch komplexer Busmodelle an, weil sich die Laufzeitperformance durch Einsatz von Parallelisierungskonzepten enorm verbessern lässt. Denkbar ist beispielsweise der in [Liu09] realisierte Ansatz, bei dem die laufzeitintensive Bewertung der einzelnen Busparameterkonfigurationen parallelisiert durchgeführt wird.

Der Abstand zwischen der Performance des genetischen Algorithmus und den beiden anderen Verfahren wächst mit zunehmender Komplexität der Busmodelle offensichtlich an. Vor allem die vielfältigen Operatoren und Anpassungsmöglichkeiten erschweren eine gute

Adaption des Verfahrens an gegebene Problemstellungen. Besteht die Möglichkeit spezielle Operatoren, die direkten Bezug zur Optimierungsaufgabe haben, zu erstellen und einzusetzen, kann die Performance des Algorithmus möglicherweise gesteigert werden. Dafür ist jedoch ein ungleich größerer Aufwand nötig als beispielsweise bei der Adaption der Partikel Schwarm Optimierung. Für den hier vorliegenden Anwendungsfall muss mit den eingesetzten, allgemeinen Operatoren zunächst einmal eine geringere Eignung des Verfahrens konstatiert werden.



**Abbildung 4.17:** Vergleich der Konvergenzverläufe des Simualted Annealing und der Partikel Schwarm Optimierung nach Anpassung der Algorithmen für Modell 1



**Abbildung 4.18:** Vergleich der Konvergenzverläufe des Simualted Annealing und der Partikel Schwarm Optimierung nach Anpassung der Algorithmen für Modell 2



# 5 Anwendung des Analyseverfahrens

Nach den algorithmentheoretischen Untersuchungen des vorangegangenen Kapitels, deren Ziel eine Bewertung der Einsetzbarkeit unterschiedlicher Optimierungsverfahren zur Steuerung der Parametervariation innerhalb der Bussystemanalysen war, erfolgt in diesem Kapitel nun eine kurze exemplarische Analyse, die den praktischen Nutzen der Anwendung verdeutlichen soll. Zu diesem Zweck wird ein neues Testmodell erstellt (siehe Abbildung 5.1). Das Modell verfügt über fünf Komponenten, die über ein gemeinsames Busnetzwerk, das einen passiven Stern beinhaltet, verbunden sind. Die einzelnen Verbindungsleitungen haben die folgende Länge:

- Bus 1: 5 m
- Bus 2: 10 m
- Bus 3: 5 m
- Bus 4: 5 m
- Bus 5: 15 m

Abgeschlossen werden die einzelnen Zweige durch Split-Terminierungen mit den Werten für die einzelnen Widerstände der Terminierung an ECU 1 und ECU 4 von  $R = 47,5 \Omega$  und  $R = 1,3 \text{ k}\Omega$  für ECU 2, ECU 3 und ECU 5.

Als Sender für die Analyse des Busmodells arbeitet ECU 1. Durch den gegebenen Aufbau der Bustopologie muss erwartet werden, dass es zu einer starken Beeinträchtigung der Signalintegrität auf dem Bus kommt. Praktische Erfahrungen haben gezeigt, dass es bei dem in dieser Topologie verwendeten Verhältnis der Längen der Busleitungen zueinander höchstwahrscheinlich zu starken Reflexionen auf dem Bus kommt. Diese überlagern die eigentlichen Signale und können zu Schwierigkeiten beim Empfang führen. Das Ziel der Untersuchung dieser, aufgrund praktischer Erfahrungen als kritisch eingestuften Topologie ist es, simulatorisch zu überprüfen, ob die Topologie auch unter der Berücksichtigung von Schwankungen mehrerer Parameter möglicherweise noch funktionsfähig sein kann.

Es werden bei der Signalintegritätsanalyse des Modells die folgenden Parameter variiert:

- Der Wert eines Terminierungswiderstands wird innerhalb eines  $\pm 10\%$ -Intervalls um den definierten Vorgabewert variiert. Dabei wird jeder der beiden Widerstände der Split-Terminierung einzeln verändert.
- Der Abstand der Leitungen zur Massefläche wird ebenfalls variiert. Dabei werden die Busleitungen in jeweils 2,5 m lange Teilstücke unterteilt. Für jedes dieser Teilstücke wird der Abstand der Leitung zur Massefläche separat variiert. Ausgehend von einem initialen Wert von 0,1 m wird die Höhe der Leitungsabschnitte dabei in einem Wertebereich zwischen 0,01 m und 0,15 m variiert.

Es ergeben sich folglich 26 variable Parameter innerhalb des Busmodells.

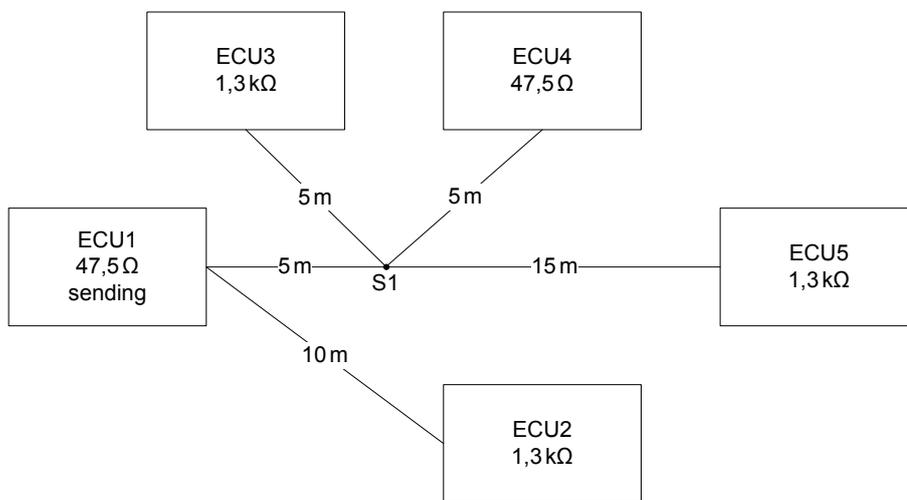


Abbildung 5.1: Topologie des Testmodells

Zu der aus Kapitel 2.2 bekannten und während der Analysen in Kapitel 4 verwendeten MSE-Bewertung wird zusätzlich eine neu implementierte Bewertungsfunktion eingesetzt.

## 5.1 Bewertung mittels Augendiagramm

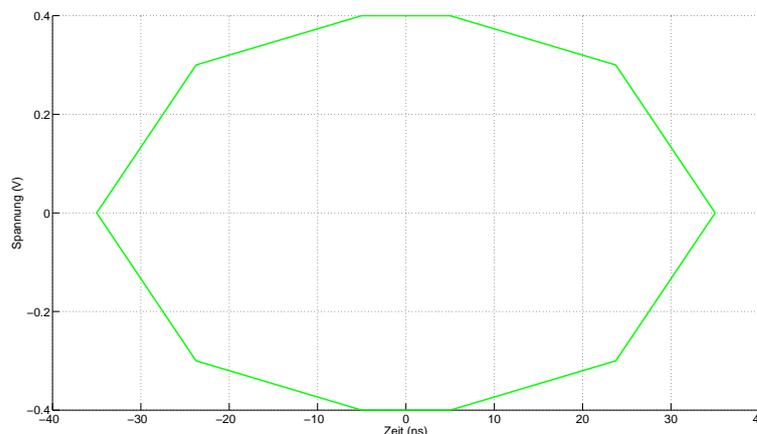
Bisher wurde die Bewertung der Signalintegrität anhand der Abweichungen zwischen dem jeweils angepassten Referenzsignal und dem vom einzelnen Transceiver empfangenen Signal bestimmt (siehe Abschnitt 2.2.4). Eine weitere Bewertungsfunktion setzt zur Bewertung der Signalintegrität ein sogenanntes Augendiagramm ein.

Bei einem Augendiagramm handelt es sich um ein vielfach eingesetztes Hilfsmittel, das zur Bewertung der Signalqualität einer digitalen Datenübertragung verwendet wird [Nus09]. Dabei wird das empfangene Signal, entsprechend der Länge eines einzelnen Informationsbits, in einzelne Abschnitte unterteilt. Bei einer messtechnischen Analyse werden diese

Signalabschnitte differenziert nach ihrem logischen Informationsgehalt (logische „1“ oder logische „0“) mittels eines Oszilloskops übereinander gelegt. Dadurch dass die logischen Pegel bei der Übertragung verschliffen werden, verschleifen die Rechteckimpulse zu verrundeten Signalen. Durch das Übereinanderlegen der Signalabschnitte ergibt sich eine Form, die einem Auge ähnelt. Bei einer guten Signalqualität ist dieses Auge deutlich geöffnet. Verschlechtert sich die Signalqualität, verkleinert sich die Öffnung des Auges zunehmend [Kar05].

Für die Auswertung der aus den Busanalysen gewonnen Signalverläufe muss eine Berechnungsgrundlage erstellt werden, die ein Maß für den Öffnungsgrad des Auges liefert. Dazu wird zunächst eine Maske definiert, die eine Grenze im Signalverlauf spezifiziert, die von 1-Bits (0-Bits) nicht unterschritten (überschritten) werden darf (siehe Abbildung 5.2).

Wird die Maske von den einzelnen Signalabschnitten geschnitten, so wird die Größe der Überschneidung als Fehlermaß genommen. Dazu werden an den Messpunkten der Signale die Abstände zwischen Signal und Maske bestimmt. Die so berechneten Überschneidungen werden über den gesamten Signalverlauf summiert und bilden so ein Maß für die Größe der Gesamtüberschneidung zwischen den Signalen und der vorgegebenen Augenmaske.



**Abbildung 5.2:** Maske zur Bestimmung der Signalintegrität mittels Augendiagramm

Die größte Überschneidung, die aus der Analyse aller zu bewertenden Signale ermittelt wird, definiert analog zur MSE-Bewertung eine Qualitätsmaßzahl der Signalintegrität des Bussystems. Das Ziel einer Optimierung ist es, wie bisher die größtmögliche Qualitätsmaßzahl zu erreichen, um die Busparameterkonfiguration zu finden, die zur größten Beeinträchtigung der Signalintegrität des Bussystems führt.

Sollte bei der Analyse eines Signals keinerlei Überschneidung zwischen dem Signalverlauf und der Maske existieren, wird für dieses eine Signal die Qualität mittels der MSE-

Bewertung bestimmt. Durch eine Gewichtung mit dem Faktor 10 für die Ergebnisse der Augendiagrammanalyse wird erreicht, dass ein Algorithmus stets die Bewertungen aus der Augendiagrammanalyse für seine Optimierungen priorisiert, falls mindestens eine solche Bewertung vorliegt.

## 5.2 Signalintegritätsanalyse

Die Analyse des gegebenen Bussystems wird mittel der in Kapitel 4 angepassten Algorithmen durchgeführt. Im Folgenden werden die Ergebnisse, die durch den Simulated Annealing Algorithmus bestimmt wurden, detailliert vorgestellt. Aus den Analysen ergeben sich die in Tabelle 5.1 dargestellten Fehlerwerte für die Signalintegrität des Bussystems. Dabei zeigen sich in allen Optimierungsläufen für ECU 1 die geringste Fehlerbewertung, für ECU 2 hingegen die stärkste Beeinträchtigung der empfangenen Signale.

Algorithmus	SA		GA		PSO	
Bewertungsfunktion	MSE	Augendiag.	MSE	Augendiag.	MSE	Augendiag.
<b>maxError</b>	2,959458	1801,055284	2,878409	1740,283745	2,930029	1761,442031
<b>ErrorECU1</b>	0,114610	25,140767	0,110086	25,008567	0,112220	24,401867
<b>ErrorECU2</b>	2,959458	1801,055284	2,878409	1740,283745	2,930029	1761,442031
<b>ErrorECU3</b>	0,459366	257,833576	0,452109	263,705149	0,469539	260,127330
<b>ErrorECU4</b>	0,621132	1352,940050	0,613585	1339,574306	0,622417	1325,174411
<b>ErrorECU5</b>	1,744482	1278,804353	1,675323	1260,001537	1,696613	1212,853927

**Tabelle 5.1:** Vergleich der ermittelten Fehlerwerte aus den Busanalysen

Die Abbildungen 5.3 und 5.4 zeigen jeweils einen Ausschnitt aus den Signalverläufen, die sich aus den Optimierungen auf Basis der beiden Bewertungsfunktionen für ECU 1 und ECU 2 ergeben, im Vergleich zu dem Signalverlauf mit der initialen Parameterkonfiguration des Bussystems. Eine Darstellung des Gesamtsignalverlaufs für die beiden ECUs findet sich im Anhang in den Abbildungen D.3 und D.4. In beiden Abbildungen sind die zum Teil sehr großen Abweichungen der empfangene Signale vom jeweils angepassten Referenzsignal gut erkennbar. Es ist auch direkt ersichtlich, dass die Abweichungen bei ECU 2 wesentlich größer sind als bei ECU 1, was folglich zu den unterschiedlichen Fehlerbewertungen führt. Beim Vergleich der an den einzelnen ECUs empfangenen Signale untereinander ist zu erkennen, dass die Optimierung auf Basis des MSE-Bewertung zu einer Erhöhung der Signalamplituden führt. Auf diese Weise ist offensichtlich eine stärkere Abweichung in den entsprechend gewichteten Signalbereichen zu erreichen (siehe Abschnitt 2.2.4). Wird die Augendiagramm-Analyse hingegen als Grundlage zur Bewertung der Signalintegrität des Bussystems genutzt, tendieren die Optimierungen dazu, die Parameter des Bussystems derart zu variieren, dass die Amplituden der Signale eher abgeschwächt werden. Dies führt folglich für die einzelnen Signalabschnitte zu einer größeren Überschneidung mit der vorgegebenen Maske des Augendiagramms. Dieses beschriebene Verhalten ist auch in den

Algorithmus	SA		GA		PSO	
	MSE	Augendiag.	MSE	Augendiag.	MSE	Augendiag.
<b>maxError</b>	2,959458	1801,055284	2,878409	1740,283745	2,930029	1761,442031
<b>TERM1BP:</b>	52,25	42,75	51,36	45,34	52,25	42,96
<b>TERM1BM:</b>	52,25	42,75	51,94	43,19	52,25	42,75
<b>TL_h1_1:</b>	0,083557	0,137932	0,138658	0,114735	0,133702	0,079628
<b>TL_h1_2:</b>	0,095023	0,083506	0,036604	0,058962	0,133887	0,149208
<b>TERM2BP:</b>	1430,00	1430,00	1411,08	1412,87	1366,79	1415,92
<b>TERM2BM:</b>	1430,00	1430,00	1350,49	1407,87	1401,08	1357,01
<b>TL_h2_1:</b>	0,071936	0,138439	0,062397	0,118455	0,015121	0,115990
<b>TL_h2_2:</b>	0,104847	0,100000	0,131775	0,138003	0,139734	0,150000
<b>TL_h2_3:</b>	0,095101	0,026690	0,055980	0,126879	0,091577	0,010030
<b>TL_h2_4:</b>	0,102681	0,100000	0,060544	0,078845	0,149602	0,150000
<b>TERM3BP:</b>	1170,00	1175,69	1299,79	1251,88	1382,98	1330,12
<b>TERM3BM:</b>	1174,03	1170,00	1179,76	1193,86	1334,77	1357,60
<b>TL_h3_1:</b>	0,150000	0,014159	0,075657	0,098803	0,149996	0,010373
<b>TL_h3_2:</b>	0,150000	0,013883	0,138117	0,076169	0,138113	0,042179
<b>TERM4BP:</b>	42,75	52,25	42,80	52,08	42,75	52,25
<b>TERM4BM:</b>	42,75	52,25	42,81	51,98	42,86	52,25
<b>TL_h4_1:</b>	0,077125	0,047139	0,078106	0,018963	0,138411	0,010007
<b>TL_h4_2:</b>	0,108567	0,010000	0,147457	0,128623	0,029737	0,010048
<b>TERM5BP:</b>	1421,85	1430,00	1347,48	1331,68	1296,86	1226,00
<b>TERM5BM:</b>	1376,75	1430,00	1303,77	1413,78	1327,55	1397,30
<b>TL_h5_1:</b>	0,010000	0,047893	0,041215	0,089710	0,096628	0,149764
<b>TL_h5_2:</b>	0,010000	0,028206	0,014094	0,012832	0,034273	0,010024
<b>TL_h5_3:</b>	0,077879	0,028924	0,010547	0,086768	0,010385	0,010052
<b>TL_h5_4:</b>	0,088396	0,013314	0,072760	0,144316	0,100316	0,010215
<b>TL_h5_5:</b>	0,138846	0,100377	0,060350	0,037418	0,018481	0,010067
<b>TL_h5_6:</b>	0,112235	0,100000	0,079673	0,148430	0,100830	0,044626
<b>Anzahl Simulationen</b>	675	521	2020	1260	1920	2020
<b>stall</b>	199	34	600	900	600	100

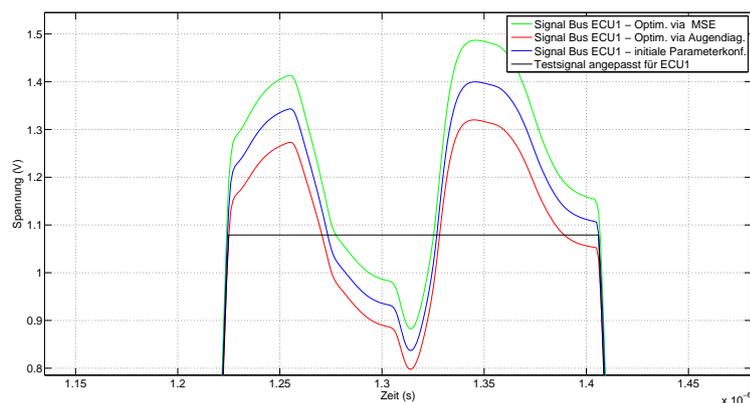
**Tabelle 5.2:** Vergleich der ermittelten Busparameterkonfigurationen mit dem größten Fehlerwert

Augendiagrammen für ECU 1 und ECU 2 erkennbar. Die Abbildungen D.1 und D.2 im Anhang dieser Arbeit zeigen die Signalverläufe mit der initialen Wertebelegung der Parameter und die Signalverläufe, die sich durch die Parameterwerte ergeben, die mittels Optimierung auf Basis der Augendiagramm-Analyse bestimmt wurden. Generell lässt sich aber erkennen, dass die hier eingesetzten variablen Parameter des Busmodells relativ geringen Einfluss auf die Signalintegrität des Bussystems haben. Um stärkere Auswirkungen zu erzielen, ist die Berücksichtigung zusätzlicher beziehungsweise anderer Variationsparameter für das Busmodell notwendig. Hier bietet sich beispielsweise eine Berücksichtigung von unterschiedlichen Temperaturen für die einzelnen ECUs oder eine Berücksichtigung von Schwankungen in der Versorgungsspannung der Bus-Komponenten an. Diese sind mit den aktuell verfügbaren Komponentenmodellen allerdings noch nicht realisierbar.

Die Parameterkonfigurationen die durch die Analyseumgebung aus den einzelnen Läufen jeweils ermittelt wurden, werden in Tabelle 5.2 differenziert nach Algorithmen und je-

weiliger Bewertungsfunktion dargestellt. In der Tabelle bezeichnet  $TermX$  den Wert der Terminierungswiderstände an ECUX (angegeben in Ohm),  $TL_hX_$  die Höhe der einzelnen Leitungsabschnitte von BusX (angegeben in Metern). In der Tabelle sind auch die Simulationszahlen enthalten bis die einzelne Analyse jeweils terminiert. Die Zahl der Simulationen, nach denen das Optimum im jeweiligen Lauf bestimmt wurde, ergibt sich aus der Differenz der angegebenen Gesamtzahl der Simulationen und des *stall*-Wertes, der beziffert, über wie viele Simulationen während der Analyse, vor der Terminierung, keine weiteren schlechteren Parameterkonfigurationen mehr gefunden wurden.

Die Analyse, die den Simulated Annealing Algorithmus eingesetzt hat, zeigt auch hier wieder das performanteste Verhalten. Die Analysen mittels GA und PSO benötigen deutlich mehr Simulationen. Die Ergebnisse der einzelnen Bussystem-Analysen zeigen, wie bereits aus den ermittelten Fehlerwerten zu vermuten war, nur geringe Unterschiede. Einzig die Differenzen, die, resultierend aus den beiden unterschiedlichen Bewertungsfunktionen, schon an den Signalplots erkennbar waren, sind auch in den ermittelten Parameterwerten ersichtlich.



**Abbildung 5.3:** Ausschnitt aus den Signalverläufen an ECU 1

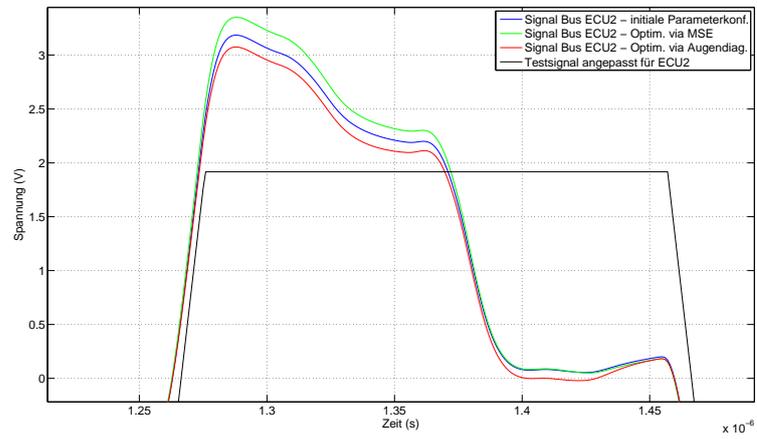


Abbildung 5.4: Ausschnitt aus den Signalverläufen an ECU 2



## 6 Zusammenfassung und Ausblick

In heutigen Kraftfahrzeugen wird aufgrund stetig steigender und sich verändernder Anforderungen eine große Zahl elektronischer Systeme verbaut, um diesen gestiegenen Anforderungen gerecht zu werden. Zwischen den verschiedenen Systemen, welche in die Fahrzeuge integriert werden, werden untereinander viele Informationen ausgetauscht. Zusätzliche Informationen versetzen einzelne Systeme in die Lage, ihre Leistungsfähigkeit zu verbessern. Aufgrund der hohen Datenmenge, die aus dem Einsatz moderner, elektronischer Systeme und ihrer Kommunikation untereinander resultieren, bedarf es einer geeigneten Kommunikationsinfrastruktur innerhalb des Fahrzeugs. Eine solche Infrastruktur stellen Kfz-Bussysteme dar.

Da Bussysteme auch sicherheitskritische Daten zwischen den verbauten Systemen austauschen, werden an die Signalintegrität der verwendeten Busaufbauten besonders hohe Anforderungen gestellt. Durch Parametervariation einzelner Komponenten, z. B. Fertigungstoleranzen von verbauten Komponenten, kann es zu Beeinträchtigung der Signalintegrität auf dem Datenbus kommen. Deshalb ist eine frühzeitige Untersuchung der Robustheit des Busaufbaus notwendig. Problematisch erweist sich dabei, dass möglicherweise erst die Kombination mehrerer, gleichzeitiger Parametervariationen zu kritischen Gesamtkonfigurationen für die Signalintegrität des Busses führen. Aufgrund der Vielzahl möglicher variabler Parameter des Bussystems ist eine automatisierte Analyse mittels Simulation ein geeignetes Mittel die Signalintegrität eines Busaufbaus zu bewerten.

Die vorliegende Arbeit befasst sich mit der Steuerung der Parametervariation während der automatisierten Bus-Analyse. Dazu wurden unterschiedliche Verfahren ausgewählt, implementiert und ihre Leistung bewertet. Da es sich bei den implementierten Verfahren um heuristische Ansätze zur Optimierung handelt, wurden im Laufe der Arbeit umfangreiche Tests durchgeführt, die eine möglichst gute Anpassung der Algorithmen gewährleisten sollen. Bei diesen Tests wurden sowohl die allgemeinen Algorithmusparameter adaptiert, als auch einige Verfahrensänderungen an den ursprünglichen Algorithmen vorgenommen.

Im Verlaufe der Arbeit wurde gezeigt, dass sich zwei der Algorithmen grundsätzlich für den Einsatz innerhalb der Bussystemanalyse eignen. Diese Einschätzung gründet sich auf den erreichten Ergebnissen und den benötigten Laufzeiten der Algorithmen. Eine Empfeh-

lung für den Einsatz eines bestimmten Algorithmus ist von der Komplexität der Modelle abhängig. Während bei kleinen Modellen die Optimierungen aufgrund seiner geringen Laufzeit mittels Simulated Annealing Vorteile gezeigt haben, können durch den Einsatz der parallel suchenden Partikel Schwarm Optimierung erst bei sehr großen Busmodellen Vorteile gegenüber dem Simulated Annealing erzielt werden. Nach der erfolgten Anpassung der Algorithmen wurden diese an einem praktischen Beispiel eingesetzt. Dabei wurde eine zusätzliche Bewertungsfunktion, die Augendiagramm-Analyse, zur Bewertung der Signalintegrität des gegebenen Bussystems implementiert und die Ergebnisse im Vergleich zur MSE-Bewertung betrachtet. Die im Busmodell realisierten Variationsparameter (Wert der Terminierungswiderstände und Abstand der Leitungen zur Massefläche) zeigten relativ geringe Auswirkungen auf den Verlauf der Signale. Um die Auswirkungen von weiteren Parametern eines Bussystems zu bestimmen, ist die Erweiterung der derzeitigen Modelle der Buskomponenten nötig. Parameter, deren Variation möglicherweise den Signalverlauf stärker beeinträchtigen könnten, die Temperatur der Komponenten sowie eine Berücksichtigung von Schwankungen in der Versorgungsspannung der ECUs sein. Um die Performance der Analyse, speziell für sehr große Bussysteme weiter zu verbessern, bietet sich eine Parallelisierung der Analysemethoden an. Insbesondere der Ansatz der Partikel Schwarm Optimierung sollte mit relativ geringem Aufwand, beispielweise mittels des in [Liu09] entwickelten Konzeptes, parallelisierbar sein.

# A Weitere Ergebnisse der Anpassung des SA

Anmerkung zu den im Folgenden angegebenen Ergebnissen:

Die hier angegebenen Laufzeiten wurden auf einem Intel Xeon Quad-Core E5310 mit einer Taktfrequenz von 1,6 GHz und 8 GB Arbeitsspeicher erreicht. Die in den Tabellen aufgeführten Laufzeiten sind nur bedingt vergleichbar. Aufgrund zeitweiliger Engpässe der zur Verfügung stehenden Lizenzen für den VHDL-Simulator kam es zu Wartezeiten einzelner Simulationen während der Bussystemanalysen. Daher korrelieren die Laufzeiten nicht für jeden angegebenen Lauf mit der Zahl durchgeführter Simulationen.

<b>Nr</b>	<b>Laufzeit</b> (hh:mm:ss,000)	<b>maxMSE</b>	<b>Anzahl</b> <b>Simulationen</b>
<b>1</b>	03:47:44,446	0,725755	352
<b>2</b>	03:46:07,924	0,725755	352
<b>3</b>	03:31:12,589	0,734617	352
<b>4</b>	03:31:28,950	0,734617	352
<b>5</b>	03:32:48,298	0,725938	352
<b>6</b>	03:32:47,485	0,725938	352
<b>7</b>	03:31:10,296	0,725755	352
<b>8</b>	03:31:08,624	0,725755	352
<b>9</b>	04:14:07,913	0,725755	352
<b>10</b>	04:06:09,138	0,734617	352

**Tabelle A.1:** SA - initiale Analyse: Ergebnisse der Einzelläufe - Modell 1

Nr	Laufzeit (hh:mm:ss,000)	maxMSE	Anzahl Simulationen
1	34:57:20,702	0,154665	902
2	34:14:28,799	0,154033	902
3	34:26:31,521	0,155913	902
4	36:03:45,371	0,153903	902
5	33:35:16,933	0,154915	902
6	35:50:59,247	0,152287	902
7	34:43:35,612	0,15476	902
8	33:10:22,843	0,153095	902
9	34:07:43,919	0,154559	902
10	35:15:09,879	0,153095	902

Tabelle A.2: SA - initiale Analyse: Ergebnisse der Einzelläufe - Modell 2

Nr	Laufzeit (hh:mm:ss,000)	maxMSE	Anzahl Simulationen
1	02:13:51,236	0,714506	224
2	02:12:59,465	0,731743	224
3	02:14:10,019	0,728387	224
4	02:13:04,936	0,727146	224
5	02:15:46,700	0,705645	224
6	02:16:24,499	0,739528	224
7	02:15:17,909	0,735909	224
8	02:15:40,925	0,731351	224
9	02:14:16,631	0,709245	224
10	02:14:10,935	0,732605	224

Tabelle A.3: SA - Parameter  $\alpha = 0,7$ : Ergebnisse der Einzelläufe - Modell 1

Nr	Laufzeit (hh:mm:ss,000)	maxMSE	Anzahl Simulationen
1	07:27:26,749	0,742376	704
2	07:02:27,664	0,745363	704
3	07:02:05,755	0,748052	704
4	07:31:59,805	0,748165	704
5	07:16:04,798	0,747078	704
6	08:54:08,582	0,736908	704
7	10:26:03,333	0,743082	704
8	08:22:04,063	0,746834	704
9	07:00:02,147	0,731502	704
10	07:00:28,992	0,746438	704

Tabelle A.4: SA - Parameter  $\alpha = 0,9$ : Ergebnisse der Einzelläufe - Modell 1

Nr	$\alpha$	Laufzeit (hh:mm:ss,000)	maxMSE	Anzahl Simulationen
1	0,7	21:12:56,803	0,146482	574
2	0,7	21:23:20,821	0,151989	574
3	0,7	21:32:14,671	0,153859	574
4	0,7	21:16:23,629	0,144962	574
5	0,7	22:48:36,676	0,146482	574
6	0,7	21:18:54,376	0,144175	574
7	0,7	22:38:22,842	0,151989	574
8	0,7	22:53:50,305	0,153859	574
9	0,7	21:50:37,515	0,152964	574
10	0,7	22:24:07,919	0,144962	574

**Tabelle A.5:** SA - Parameter  $\alpha = 0,7$ : Ergebnisse der Einzelläufe - Modell 2

Nr	$\alpha$	Laufzeit (hh:mm:ss,000)	maxMSE	Anzahl Simulationen
1	0,9	69:11:30,766	0,156037	1804
2	0,9	70:22:47,008	0,155369	1804
3	0,9	69:53:14,866	0,156541	1804
4	0,9	68:43:37,834	0,155997	1804
5	0,9	67:18:38,738	0,156559	1804
6	0,9	71:21:27,084	0,156116	1804
7	0,9	71:27:55,798	0,157734	1804
8	0,9	67:29:16,734	0,157365	1804
9	0,9	68:48:31,148	0,156113	1804
10	0,9	68:46:12,574	0,155309	1804

**Tabelle A.6:** SA - Parameter  $\alpha = 0,9$ : Ergebnisse der Einzelläufe - Modell 2

<b>Nr</b>	<b><math>\alpha</math></b>	<b>Laufzeit</b> (hh:mm:ss,000)	<b>maxMSE</b>	<b>Anzahl</b> <b>Simulationen</b>
1	0,7	03:01:59,845	0,752009	302
2	0,7	02:58:22,412	0,754399	301
3	0,7	02:46:14,786	0,749426	280
4	0,7	02:56:03,027	0,754396	293
5	0,7	03:01:44,508	0,731334	302
6	0,7	02:59:20,696	0,752669	300
7	0,7	02:48:24,106	0,744388	283
8	0,7	02:53:21,447	0,753218	291
9	0,7	03:03:40,862	0,753721	308
10	0,7	02:55:23,570	0,745299	290
11	0,8	04:08:04,727	0,754342	395
12	0,8	04:08:29,032	0,751529	294
13	0,8	04:12:42,879	0,755006	443
14	0,8	03:51:01,361	0,754248	454
15	0,8	04:02:13,495	0,753048	438
16	0,8	04:07:13,205	0,754378	446
17	0,8	02:41:43,638	0,740617	420
18	0,8	03:23:34,484	0,744419	450
19	0,8	02:51:18,780	0,744211	329
20	0,8	04:19:27,551	0,754737	359
21	0,9	08:49:28,709	0,754199	842
22	0,9	08:36:57,460	0,754144	857
23	0,9	05:15:23,696	0,753648	528
24	0,9	09:10:09,974	0,753111	863
25	0,9	05:44:16,899	0,754593	573
26	0,9	09:49:27,132	0,754199	842
27	0,9	08:32:09,298	0,754144	857
28	0,9	05:12:16,579	0,753648	528
29	0,9	08:29:40,336	0,753111	863
30	0,9	05:39:10,789	0,754593	573

**Tabelle A.7:** SA - Lokalisierungsschritte, mit Schrittweite 2,5 %:  
Ergebnisse der Einzelläufe - Modell 1

<b>Nr</b>	<b><math>\alpha</math></b>	<b>Laufzeit</b> (hh:mm:ss,000)	<b>maxMSE</b>	<b>Anzahl</b> <b>Simulationen</b>
<b>1</b>	0,7	02:51:44,750	0,752607	287
<b>2</b>	0,7	02:46:59,212	0,75205	278
<b>3</b>	0,7	02:41:57,308	0,743560	272
<b>4</b>	0,7	02:51:36,501	0,750358	287
<b>5</b>	0,7	02:33:49,286	0,744246	259
<b>6</b>	0,7	02:47:27,056	0,752444	281
<b>7</b>	0,7	03:01:12,099	0,754035	303
<b>8</b>	0,7	02:56:57,998	0,754607	294
<b>9</b>	0,7	02:08:25,036	0,753110	216
<b>10</b>	0,7	02:39:47,281	0,744155	270
<b>11</b>	0,8	04:31:14,147	0,754930	438
<b>12</b>	0,8	04:16:31,535	0,752452	426
<b>13</b>	0,8	04:14:29,869	0,751135	425
<b>14</b>	0,8	02:49:44,553	0,754098	282
<b>15</b>	0,8	04:18:19,281	0,752925	433
<b>16</b>	0,8	04:24:37,802	0,754930	438
<b>17</b>	0,8	04:17:46,969	0,752452	426
<b>18</b>	0,8	04:15:10,440	0,751135	425
<b>19</b>	0,8	02:50:30,294	0,754098	282
<b>20</b>	0,8	04:21:21,254	0,752925	433
<b>21</b>	0,9	08:13:00,680	0,754699	817
<b>22</b>	0,9	07:58:39,731	0,754747	798
<b>23</b>	0,9	04:41:06,917	0,744066	474
<b>24</b>	0,9	08:14:27,988	0,753583	816
<b>25</b>	0,9	08:15:39,463	0,754310	832
<b>26</b>	0,9	04:45:56,866	0,753596	477
<b>27</b>	0,9	08:17:49,004	0,753643	833
<b>28</b>	0,9	08:08:11,017	0,754107	819
<b>29</b>	0,9	07:53:03,710	0,754298	792
<b>30</b>	0,9	08:10:17,324	0,754277	824

**Tabelle A.8:** SA - Lokalisierungsschritte, mit Schrittweite 5 %:  
Ergebnisse der Einzelläufe - Modell 1

Nr	$\alpha$	Laufzeit (hh:mm:ss,000)	maxMSE	Anzahl Simulationen
1	0,7	02:06:46,901	0,754533	211
2	0,7	03:37:29,931	0,754106	283
3	0,7	04:18:39,643	0,754933	288
4	0,7	04:05:11,033	0,753607	275
5	0,7	04:09:57,295	0,754674	282
6	0,7	03:22:49,732	0,748203	240
7	0,7	03:27:01,181	0,753222	286
8	0,7	02:52:48,289	0,754866	272
9	0,7	02:42:48,039	0,753962	270
10	0,7	02:47:43,465	0,75246	281
11	0,8	04:08:04,727	0,754342	418
12	0,8	04:08:29,032	0,751529	422
13	0,8	04:12:42,879	0,755006	426
14	0,8	03:51:01,361	0,754248	389
15	0,8	04:02:13,495	0,753048	415
16	0,8	04:07:13,205	0,754378	417
17	0,8	02:41:43,638	0,740617	273
18	0,8	03:23:34,484	0,744419	344
19	0,8	02:51:18,780	0,744211	289
20	0,8	04:19:27,551	0,754737	435
21	0,9	04:05:37,753	0,751491	415
22	0,9	07:40:41,942	0,7532	782
23	0,9	07:51:55,697	0,754383	797
24	0,9	07:46:00,006	0,754652	785
25	0,9	04:24:33,938	0,753237	450
26	0,9	07:48:55,372	0,754187	791
27	0,9	07:47:31,678	0,754171	784
28	0,9	07:39:32,223	0,754176	641
29	0,9	09:14:36,137	0,749814	637
30	0,9	06:25:15,184	0,752898	452

**Tabelle A.9:** SA - Lokalisierungsschritte, mit Schrittweite 10 %:  
Ergebnisse der Einzelläufe - Modell 1

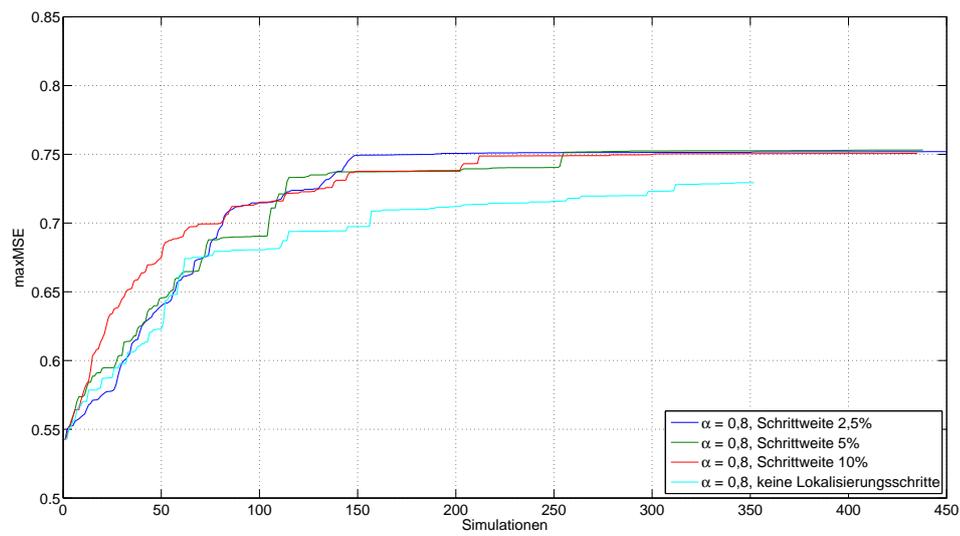


Abbildung A.1: SA - Lokalisierungsschritte für  $\alpha = 0,8$ : Konvergenzverhalten - Modell 1

Nr	alpha	Laufzeit (hh:mm:ss,000)	MSE	Anzahl Simulationen
1	0,7	24:40:23,853	0,157064	637
2	0,7	33:50:32,363	0,157884	866
3	0,7	29:22:36,028	0,157700	707
4	0,7	34:27:28,606	0,157884	866
5	0,7	28:08:25,329	0,157700	707
6	0,8	48:01:44,585	0,158655	1247
7	0,8	49:10:55,343	0,157250	1330
8	0,8	46:44:24,276	0,158160	1225
9	0,8	50:28:45,072	0,157664	1279
10	0,8	48:07:33,561	0,157196	1223
11	0,9	51:04:47,350	0,157109	1321
12	0,9	52:00:16,868	0,157417	1351
13	0,9	69:02:02,357	0,157101	1758
14	0,9	95:00:30,173	0,157217	2481
15	0,9	74:35:44,679	0,159610	1888

Tabelle A.10: SA - Lokalisierungsschritte, mit Schrittweite 2,5 %:  
Ergebnisse der Einzelläufe - Modell 2

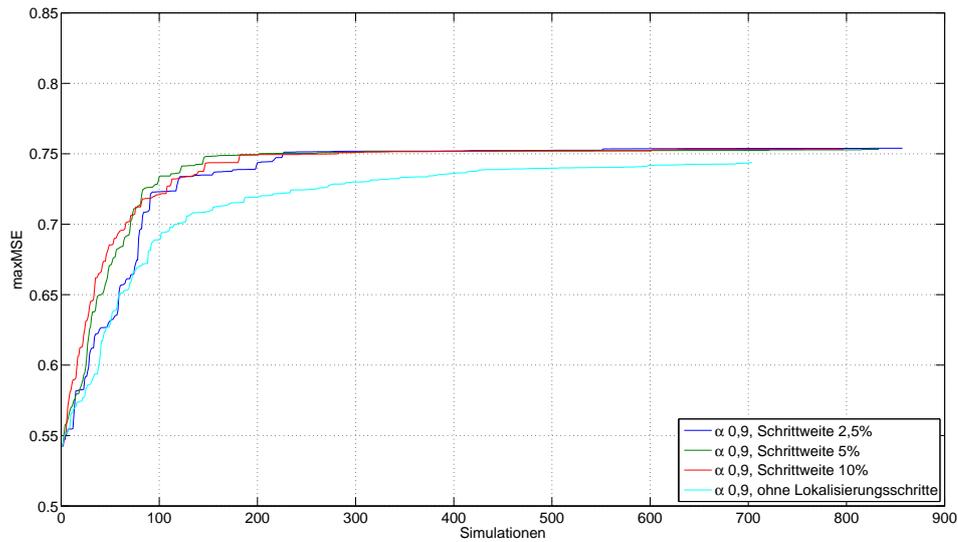


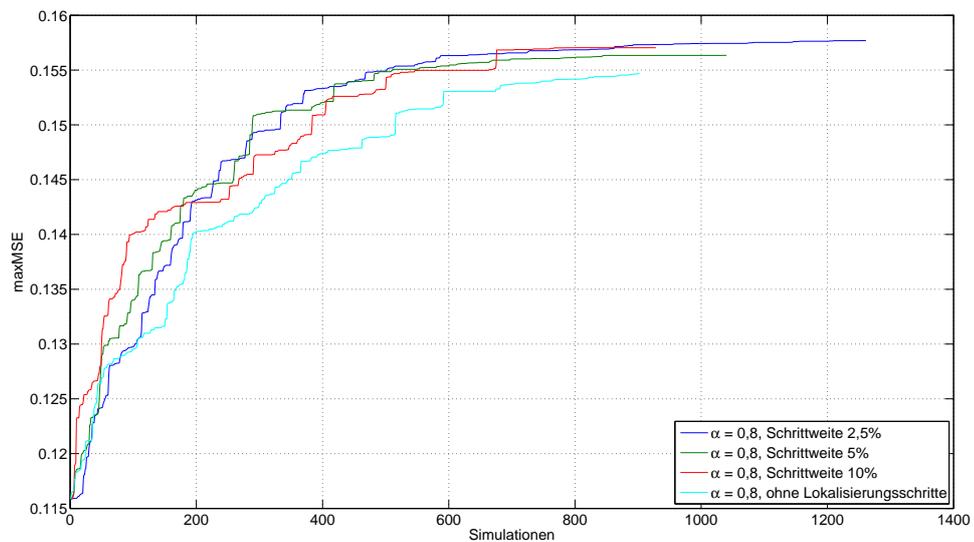
Abbildung A.2: SA - Lokalisierungsschritte für  $\alpha = 0,9$ : Konvergenzverhalten - Modell 1

Nr	alpha	Laufzeit (hh:mm:ss,000)	MSE	Anzahl Simulationen
1	0,7	31:48:45,188	0,157233	808
2	0,7	23:43:16,700	0,157226	608
3	0,7	25:50:13,530	0,15809	659
4	0,7	23:58:37,018	0,157226	608
5	0,7	25:55:13,000	0,15809	659
6	0,8	47:30:46,841	0,156572	1235
7	0,8	27:41:21,217	0,155519	750
8	0,8	40:51:8,471	0,157161	1075
9	0,8	38:40:55,671	0,15718	969
10	0,8	45:24:37,327	0,156323	1171
11	0,9	40:38:21,207	0,158463	1097
12	0,9	63:37:41,736	0,156506	1653
13	0,9	93:37:38,900	0,156905	2407
14	0,9	92:48:35,455	0,157857	2412
15	0,9	42:58:52,998	0,158463	1097

Tabelle A.11: SA - Lokalisierungsschritte, mit Schrittweite 5 %:  
Ergebnisse der Einzelläufe - Modell 2

Nr	alpha	Laufzeit (hh:mm:ss,000)	MSE	Anzahl Simulationen
1	0,7	29:30:39,183	0,159798	747
2	0,7	28:06:31,497	0,157908	702
3	0,7	30:15:56,377	0,158996	772
4	0,7	28:20:58,077	0,157338	742
5	0,7	28:38:20,664	0,156425	758
6	0,8	30:53:50,088	0,158191	780
7	0,8	37:43:33,668	0,156697	953
8	0,8	36:50:41,340	0,156667	958
9	0,8	28:57:42,702	0,158191	780
10	0,8	44:17:55,745	0,156381	1171
11	0,9	86:07:07,566	0,159671	2313
12	0,9	76:23:14,111	0,158133	2064
13	0,9	62:37:55,173	0,155971	1660
14	0,9	64:17:09,585	0,156838	1757
15	0,9	85:31:33,894	0,159671	2313

**Tabelle A.12:** SA - Lokalisierungsschritte, mit Schrittweite 10 %:  
Ergebnisse der Einzelläufe - Modell 2



**Abbildung A.3:** SA - Lokalisierungsschritte für  $\alpha = 0,8$ : Konvergenzverhalten - Modell 2

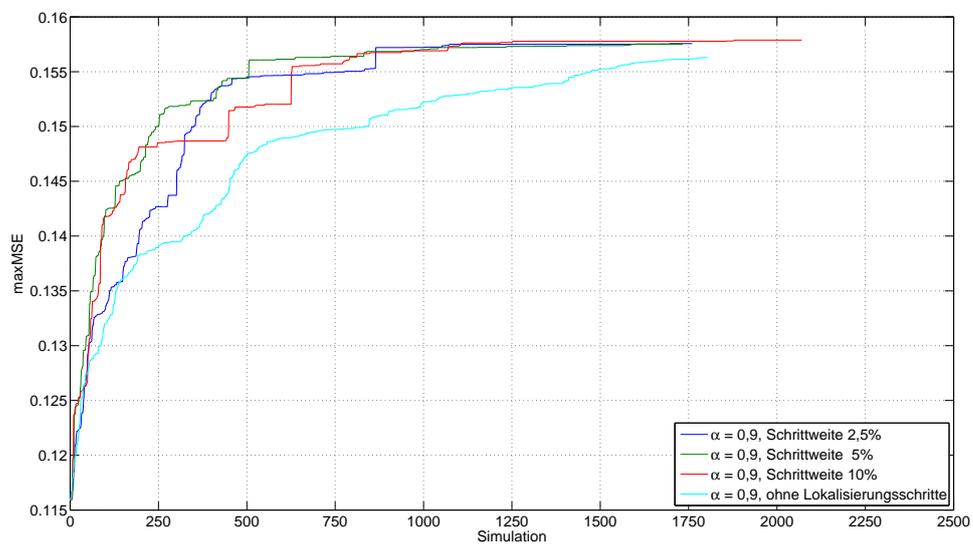


Abbildung A.4: SA - Lokalisierungsschritte für  $\alpha = 0,9$ : Konvergenzverhalten - Modell2

# B Weitere Ergebnisse der Anpassung des GA

Anmerkung zu den im Folgenden angegebenen Ergebnissen:

Die hier angegebenen Laufzeiten wurden auf einem Intel Xeon Quad-Core E5310 mit einer Taktfrequenz von 1,6 GHz und 8 GB Arbeitsspeicher erreicht. Die in den Tabellen aufgeführten Laufzeiten sind nur bedingt vergleichbar. Aufgrund zeitweiliger Engpässe der zur Verfügung stehenden Lizenzen für den VHDL-Simulator kam es zu Wartezeiten einzelner Simulationen während der Bussystemanalysen. Daher korrelieren die Laufzeiten nicht für jeden angegebenen Lauf mit der Zahl durchgeführter Simulationen.

Nr	Populationsgröße	Laufzeit (hh:mm:ss,000)	MSE	Anzahl Simulationen
1	5	05:01:42,651	0,620241	505
2	5	05:55:36,583	0,625228	505
3	5	02:17:27,101	0,583424	205
4	5	05:33:03,354	0,619299	475
5	5	03:42:12,520	0,664456	360
6	5	02:00:37,868	0,638006	205
7	5	05:10:59,836	0,627271	505
8	5	06:02:01,053	0,663173	505
9	5	05:47:29,652	0,625948	505
10	5	05:33:20,115	0,640435	505
11	10	10:05:26,987	0,662141	1010
12	10	04:50:24,393	0,622059	480
13	10	08:29:27,448	0,687246	850
14	10	08:18:21,808	0,721609	830
15	10	08:42:06,201	0,725689	880
16	10	04:57:35,070	0,649915	470
17	10	10:00:43,458	0,645011	1010
18	10	10:02:06,292	0,677009	1010
19	10	09:59:59,145	0,703475	1010
20	10	07:15:27,265	0,618852	710
21	15	15:02:13,176	0,691057	1515
22	15	12:01:46,722	0,679714	1185
23	15	15:25:43,380	0,718022	1515
24	15	16:30:32,663	0,684018	1515
25	15	17:10:42,055	0,691809	1515
26	15	14:55:16,767	0,734978	1515

27	15	15:21:12,789	0,742096	1515
28	15	15:14:05,772	0,739394	1515
29	15	14:56:15,941	0,741511	1515
30	15	14:49:58,237	0,707365	1515
31	20	17:21:03,863	0,722271	1740
32	20	24:45:36,105	0,740172	2020
33	20	27:58:12,747	0,721712	2020
34	20	39:45:03,569	0,739516	2020
35	20	39:25:00,340	0,692117	2020
36	20	44:04:12,471	0,737995	2020
37	20	48:10:07,225	0,744377	2020
38	20	48:44:00,332	0,722200	2020
39	20	47:59:26,150	0,736265	2020
40	20	97:35:59,763	0,734947	2020
41	25	44:40:55,000	0,739944	2525
42	25	42:49:52,195	0,727082	2525
43	25	26:53:07,758	0,731138	2525
44	25	27:41:52,346	0,719185	2525
45	25	28:15:21,307	0,741192	2525
46	25	27:30:01,023	0,727082	2525
47	25	28:04:45,125	0,731514	2525
48	25	27:46:10,480	0,739944	2525
48	25	28:40:08,242	0,718505	2525
50	25	23:51:53,472	0,71545	2150
51	30	155:08:25,158	0,717216	3030
52	30	32:26:06,065	0,750743	3030
53	30	48:27:38,502	0,736826	3030
54	30	37:35:01,197	0,738347	2940
55	30	38:50:00,613	0,735238	3030
56	30	34:21:31,636	0,709886	3030
57	30	33:28:52,549	0,745763	3030
58	30	31:46:35,082	0,740433	2940
59	30	31:16:58,068	0,749419	3030
60	30	28:39:09,348	0,713096	2580
61	35	41:11:57,900	0,745374	3535
62	35	56:59:21,476	0,743574	3535
63	35	57:04:37,383	0,744464	3535
64	35	66:52:56,746	0,729326	3535
65	35	48:10:05,534	0,750961	3535
66	35	50:51:13,281	0,729326	3535
67	35	44:09:21,451	0,740783	3535
68	35	24:20:08,000	0,744407	3535
69	35	39:57:01,702	0,745424	3535
70	35	37:15:22,455	0,729326	3535

Tabelle B.1: GA - Populationsgröße: Ergebnisse der Einzelläufe - Modell 1

<b>Nr</b>	<b>Populationsgröße</b>	<b>Laufzeit</b> (hh:mm:ss,000)	<b>MSE</b>	<b>Anzahl</b> <b>Simulationen</b>
1	5	09:35:22,822	0,118289	250
2	5	06:17:10,781	0,136894	170
3	5	18:12:04,452	0,121893	475
4	5	05:47:41,316	0,110566	160
5	5	14:52:24,064	0,123872	390
6	5	09:44:37,648	0,132975	265
7	5	18:55:30,759	0,131551	505
8	5	10:32:46,798	0,133873	275
9	5	17:27:53,833	0,130518	470
10	5	18:56:55,045	0,113433	505
11	10	109:38:45,096	0,133719	1010
12	10	39:13:12,476	0,138105	1010
13	10	37:45:22,373	0,130656	1010
14	10	16:26:02,200	0,132438	440
15	10	24:20:39,365	0,134956	650
16	10	42:41:48,893	0,133335	1010
17	10	47:39:17,870	0,137516	1010
18	10	26:08:30,046	0,130468	580
19	10	37:29:37,305	0,138265	760
20	10	44:33:19,717	0,119925	1010
21	15	73:19:05,288	0,13628	1470
22	15	58:33:31,596	0,14149	1185
23	15	76:21:50,769	0,134448	1095
24	15	46:02:39,943	0,141431	1110
25	15	54:03:04,605	0,140776	1260
26	15	31:06:26,605	0,141158	780
27	15	60:15:06,322	0,139182	1515
28	15	61:06:23,858	0,135949	1515
29	15	42:53:28,515	0,141431	1110
30	15	57:02:58,032	0,139281	1515
31	20	76:34:09,204	0,142587	2020
32	20	68:12:02,923	0,140084	1840
33	20	45:32:44,687	0,142344	1240
34	20	75:24:20,055	0,142478	2020
35	20	80:06:21,300	0,141758	2020
36	20	96:58:04,401	0,152192	2020
37	20	100:30:15,226	0,140546	2020
38	20	111:45:54,097	0,138362	2020
39	20	82:37:54,739	0,142587	2020
40	20	74:49:56,260	0,142574	2020
41	25	85:05:57,510	0,138324	2225
42	25	93:;50:50,572	0,146747	2525
43	25	93:32:19,452	0,144400	2525
44	25	103:37:15,505	0,142939	2525
45	25	104:17:13,785	0,142952	2150
46	25	125:39:11,637	0,142259	2525
47	25	182:33:03,004	0,143971	2475
48	25	99:07:09,299	0,141886	2400
49	25	71:05:30,335	0,143373	1800
50	25	99:26:30,375	0,144467	2525

51	30	120:40:56,255	0,153702	3030
52	30	120:32:01,512	0,140412	3030
53	30	120:59:35,522	0,153702	3030
54	30	115:54:09,995	0,145096	3030
55	30	113:30:27,237	0,15464	3030
56	30	74:51:46,641	0,140289	2010
57	30	90:48:47,400	0,144743	2400
58	30	113:06:56,457	0,139273	3030
59	30	114:19:17,430	0,142091	3030
60	30	115:17:04,797	0,144169	3030
61	35	131:55:51,689	0,152895	3535
62	35	131:45:34,635	0,143958	3535
63	35	117:49:27,876	0,139386	2835
64	35	171:13:11,773	0,152288	3535
65	35	151:58:38,462	0,146747	3535
66	35	138:24:13,620	0,144614	3535
67	35	138:51:26,597	0,150398	3535
68	35	135:01:42,109	0,152895	3535
69	35	137:15:00,866	0,144614	3535
70	35	137:25:23,808	0,144614	3535

Tabelle B.2: GA - Populationsgröße: Ergebnisse der Einzelläufe - Modell 2

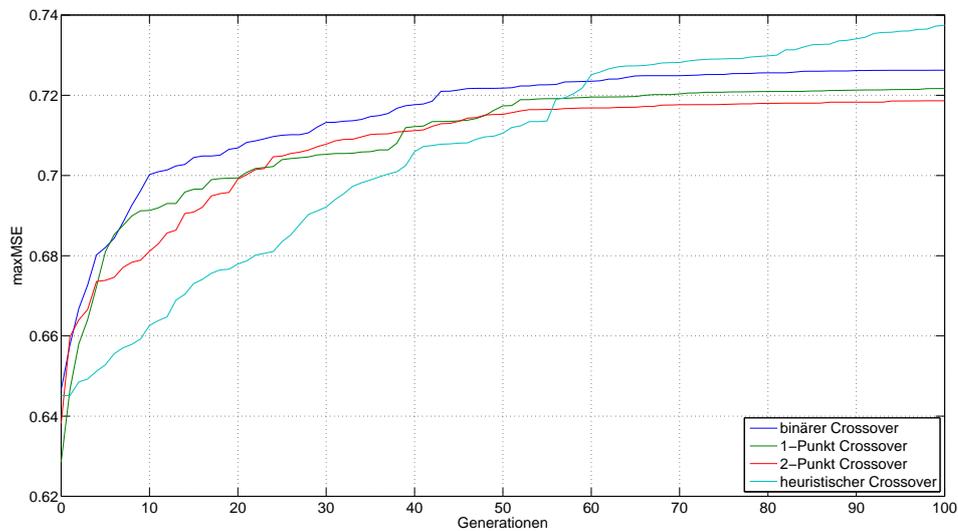


Abbildung B.1: GA - Rest-Selektion: Konvergenzverhalten - Modell 1

Nr	Crossover-Operator	Laufzeit (hh:mm:ss,000)	MSE	Anzahl Simulationen
1	1-Point	16:38:46,457	0,710273	1640
2	1-Point	21:37:24,531	0,732403	2020
3	1-Point	21:58:18,886	0,730925	2020
4	1-Point	20:16:20,157	0,742464	2020
5	1-Point	20:1:32:047	0,720474	2020
6	2-Point	20:18:57,966	0,726542	2020
7	2-Point	20:20:03,131	0,726542	2020
8	2-Point	21:21:23,718	0,729876	2020
9	2-Point	21:24:04,770	0,734043	2020
10	2-Point	20:22:33,301	0,715627	2020
11	arithmetisch	20:24:42,589	0,697975	2020
12	arithmetisch	21:55:05,616	0,618893	2020
13	arithmetisch	21:23:53,319	0,711538	2020
14	arithmetisch	19:45:48,036	0,621798	2020
15	arithmetisch	11:25:34,097	0,714369	1100
16	heuristisch(1.2)	20:27:34,343	0,739039	2020
17	heuristisch(1.2)	22:02:08,014	0,724535	2020
18	heuristisch(1.2)	21:18:51,100	0,712746	2020
19	heuristisch(1.2)	20:21:00,774	0,748795	2020
20	heuristisch(1.2)	19:58:42,466	0,744727	2020
21	intermediate(0,5)	20:22:41,336	0,701097	2020
22	intermediate(0,5)	20:02:48,000	0,654296	2020
23	intermediate(0,5)	21:01:40,801	0,685805	2020
24	intermediate(0,5)	20:17:17,657	0,663578	2020
25	intermediate(0,5)	20:14:00,203	0,667541	2020
26	binär	17:21:03,863	0,722271	1740
27	binär	24:45:36,105	0,740172	2020
28	binär	27:58:12,747	0,721712	2020
29	binär	39:45:03,569	0,739516	2020
30	binär	39:25:00,340	0,692117	2020

**Tabelle B.3:** GA - Crossover-Operatoren bei stochastisch uniformer Selektion - Modell 1

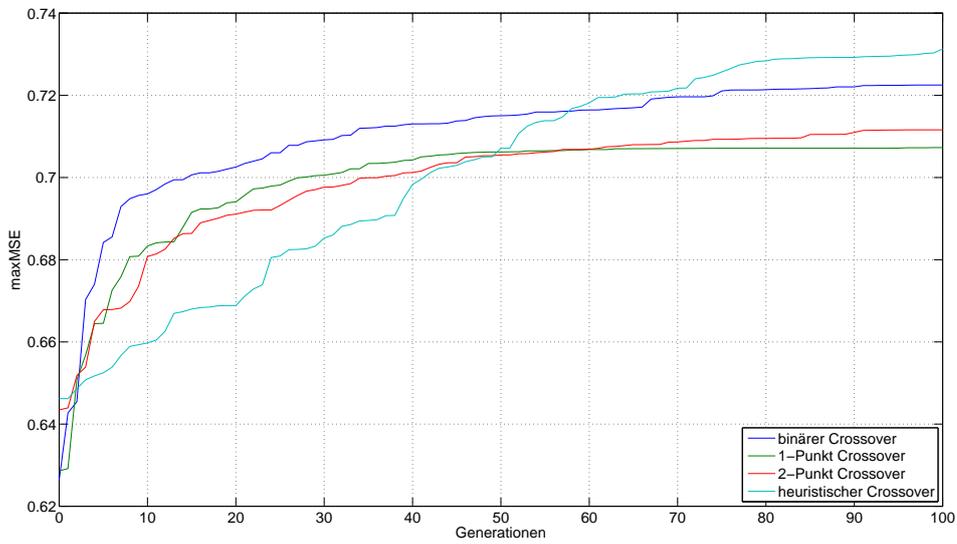


Abbildung B.2: GA - Roulette-Selektion: Konvergenzverhalten - Modell 1

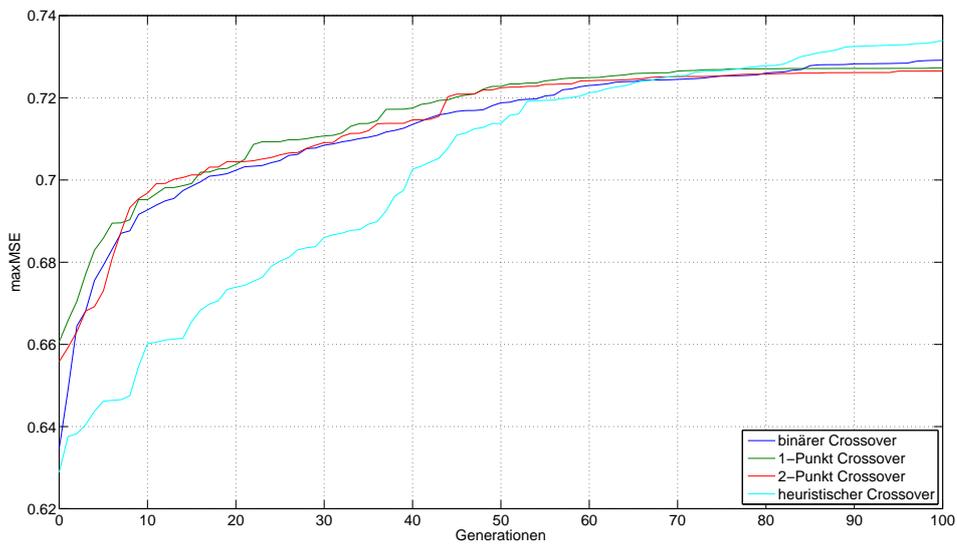


Abbildung B.3: GA - stochastisch uniforme Selektion: Konvergenzverhalten - Modell 1

Nr	Crossover:Mutation	Laufzeit (hh:mm:ss,000)	maxMSE	Anzahl Simulationen
1	20:80	20:43:40,518	0,743292	2020
2	20:80	21:40:33,499	0,741923	2020
3	20:80	21:48:01,001	0,745452	2020
4	20:80	20:50:56,019	0,734111	2020
5	20:80	20:33:39,300	0,729671	2020
6	20:80	20:13:54,764	0,730444	2020
7	20:80	21:13:51,784	0,744554	2020
8	20:80	20:04:03,160	0,746052	2020
9	20:80	19:55:17,825	0,729694	2020
10	20:80	21:34:17,120	0,747521	2020
11	40:60	21:08:55,748	0,742261	2020
12	40:60	22:08:26,612	0,734187	2020
13	40:60	21:59:08,705	0,737242	2020
14	40:60	20:55:02,844	0,748684	2020
15	40:60	20:49:26,828	0,744977	2020
16	40:60	20:55:41,064	0,742261	2020
17	40:60	21:09:26,554	0,734187	2020
18	40:60	19:58:22,989	0,737242	2020
19	40:60	21:00:36,545	0,748684	2020
20	40:60	20:45:16,645	0,737476	2020
21	60:40	21:01:37,541	0,744679	2020
22	60:40	22:02:59,066	0,720119	2020
23	60:40	18:35:55,549	0,702363	1700
24	60:40	20:31:11,467	0,732686	2020
25	60:40	19:45:11,179	0,716330	2020
26	60:40	20:26:28,407	0,750852	2020
27	60:40	21:44:30,793	0,745347	2020
28	60:40	21:20:40,006	0,739580	2020
29	60:40	20:19:44,854	0,742832	2020
30	60:40	20:06:53,231	0,732373	2020
31	80:20	20:33:25,943	0,724535	2020
32	80:20	20:27:34,343	0,739039	2020
33	80:20	22:02:08,014	0,724535	2020
34	80:20	21:18:51,100	0,712746	2020
35	80:20	20:21:00,774	0,748795	2020
36	80:20	19:58:42,466	0,744727	2020
37	80:20	21:29:02,676	0,712746	2020
38	80:20	20:37:12,318	0,748795	2020
39	80:20	20:03:47,282	0,744727	2020
40	80:20	20:54:50,452	0,742551	2020
41	100:0	08:32:49,525	0,671838	780
42	100:0	09:50:21,046	0,616336	960
43	100:0	10:27:36,072	0,681606	940
44	100:0	09:09:03,054	0,634339	800
45	100:0	08:12:11,495	0,654811	820
46	100:0	11:40:48,371	0,672088	980
47	100:0	08:43:52,846	0,687955	880
48	100:0	09:31:51,647	0,653061	900
49	100:0	08:18:47,654	0,672448	820
50	100:0	09:33:00,321	0,597048	960

**Tabelle B.4:** GA - Verhältnis zwischen Crossover und Mutation: Einzelergebnisse - Modell 1



# C Weitere Ergebnisse der Anpassung der PSO

Anmerkung zu den im Folgenden angegebenen Ergebnissen:

Die hier angegebenen Laufzeiten wurden auf einem Intel Xeon Quad-Core E5310 mit einer Taktfrequenz von 1,6 GHz und 8 GB Arbeitsspeicher erreicht. Die in den Tabellen aufgeführten Laufzeiten sind nur bedingt vergleichbar. Aufgrund zeitweiliger Engpässe der zur Verfügung stehenden Lizenzen für den VHDL-Simulator kam es zu Wartezeiten einzelner Simulationen während der Bussystemanalysen. Daher korrelieren die Laufzeiten nicht für jeden angegebenen Lauf mit der Zahl durchgeführter Simulationen.

Nr	Populationsgröße	Laufzeit (hh:mm:ss,000)	MSE	Anzahl Simulationen
1	5	06:50:05,314	0,734228	505
2	5	06:31:29,375	0,750682	505
3	5	06:56:17,815	0,730519	460
4	5	03:17:56,322	0,712438	505
5	5	02:03:31,759	0,718399	255
6	5	03:39:13,009	0,714230	175
7	5	03:08:49,365	0,694521	305
8	5	04:07:02,121	0,742801	270
9	5	05:56:02,223	0,733774	350
10	5	05:59:39,102	0,737168	505
11	10	10:52:08,626	0,739014	265
12	10	04:51:45,523	0,719723	1010
13	10	06:23:07,956	0,744378	450
14	10	10:56:43,640	0,743437	590
15	10	04:24:56,653	0,725760	1010
16	10	06:02:04,589	0,718271	410
17	10	05:11:09,538	0,740149	560
18	10	07:26:53,276	0,729558	480
19	10	03:53:00,301	0,708669	690
20	10	04:50:08,923	0,716076	360
21	15	06:06:44,420	0,742587	450
22	15	07:16:30,332	0,726709	585
23	15	08:29:05,295	0,730212	690
24	15	09:54:05,620	0,735435	810
25	15	06:48:44,771	0,715130	870
26	15	07:16:45,119	0,718264	600

27	15	09:21:58,850	0,709565	645
28	15	16:19:46,584	0,747002	735
29	15	07:32:51,855	0,724664	1515
30	15	13:58:35,763	0,749490	720
31	20	20:54:20,295	0,750450	1170
32	20	09:54:59,764	0,739513	2020
33	20	27:18:51,329	0,748349	960
34	20	13:17:01,444	0,730191	2020
35	20	13:32:06,665	0,738681	880
36	20	21:03:07,999	0,723301	980
37	20	23:53:49,066	0,745716	980
38	20	19:50:15,315	0,716457	1140
39	20	28:34:43,925	0,728761	1020
40	20	22:14:00,340	0,724700	1460
41	25	15:41:11,780	0,729015	860
42	25	20:45:06,274	0,730435	1025
43	25	28:51:50,467	0,733892	1400
44	25	48:35:28,180	0,741415	1200
45	25	19:12:12,040	0,731739	2525
46	25	23:09:11,140	0,743325	975
47	25	21:25:37,398	0,724813	1200
48	25	38:04:22,754	0,740543	1000
49	25	31:50:06,056	0,742890	1525
50	25	27:05:07,199	0,734359	1550
51	30	10:39:52,308	0,729777	1700
52	30	25:11:12,086	0,745395	1050
53	30	29:03:45,533	0,727518	2100
54	30	43:55:02,008	0,743775	2010
55	30	44:54:39,251	0,739072	3030
56	30	28:53:38,820	0,739843	2130
57	30	68:52:59,185	0,747328	1500
58	30	30:53:24,529	0,733403	3030
59	30	24:18:47,310	0,737208	1560
60	30	24:14:02,999	0,736203	1200
61	35	14:32:32,005	0,722683	1740
62	35	13:55:32,094	0,741721	1435
63	35	15:02:40,694	0,747226	1295
64	35	12:48:24,119	0,745988	1260
65	35	19:46:42,862	0,736840	1260
66	35	17:57:51,803	0,731393	1960
67	35	20:50:45,311	0,743182	1785
68	35	22:59:32,366	0,733106	2065
69	35	23:21:29,185	0,738540	2275
70	35	30:51:00,000	0,735808	2310

**Tabelle C.1:** PSO - Populationsgröße:  
Ergebnisse der Einzelläufe - Modell 1

<b>Nr</b>	<b>Populationsgröße</b>	<b>Laufzeit</b> (hh:mm:ss,000)	<b>MSE</b>	<b>Anzahl</b> <b>Simulationen</b>
1	5	23:14:57,978	0,154242	505
2	5	10:48:23,651	0,141072	230
3	5	09:06:16,202	0,149232	200
4	5	11:23:48,157	0,142929	250
5	5	14:41:05,885	0,152963	330
6	5	12:36:21,776	0,144943	285
7	5	20:12:21,604	0,153582	455
8	5	13:51:26,093	0,143689	310
9	5	12:59:57,343	0,143982	290
10	5	09:26:31,374	0,140032	215
11	10	17:42:51,325	0,144404	420
12	10	23:51:11,819	0,145076	560
13	10	24:00:26,878	0,154218	580
14	10	32:44:57,869	0,153632	800
15	10	20:49:25,952	0,152784	510
16	10	18:57:17,797	0,143906	460
17	10	02:13:53,070	0,147163	490
18	10	23:40:14,236	0,153871	510
19	10	23:37:45,290	0,151795	460
20	10	23:14:25,953	0,153281	440
21	15	81:49:55,921	0,155765	1515
22	15	28:52:09,331	0,143437	525
23	15	34:45:23,446	0,151325	540
24	15	52:26:28,405	0,148625	750
25	15	43:15:51,323	0,151089	510
26	15	46:42:31,696	0,143931	540
27	15	72:44:09,346	0,153962	825
28	15	48:55:50,409	0,144163	570
29	15	54:19:58,982	0,143527	705
30	15	125:03:12,052	0,153519	1515
31	20	43:43:20,181	0,151257	980
32	20	64:38:24,840	0,155551	1300
33	20	38:23:32,725	0,151804	780
34	20	108:27:23,632	0,155315	2020
35	20	90:12:15,297	0,154952	1480
36	20	74:44:16,541	0,155225	940
37	20	75:53:39,449	0,145539	880
38	20	70:38:55,046	0,151047	1000
39	20	63:09:08,158	0,156603	920
40	20	108:29:09,249	0,156603	2020
41	25	99:22:43,359	0,156987	2525
42	25	54:43:20,916	0,156295	1375
43	25	50:19:11,389	0,152255	1125
44	25	49:22:54,755	0,152735	1075
45	25	40:49:16,743	0,154502	1325
46	25	65:40:28,905	0,145934	1350
47	25	77:12:22,068	0,151619	1500
48	25	102:16:05,473	0,155815	1950
49	25	86:40:19,969	0,152739	1650
50	25	56:46:21,169	0,152138	1100

51	30	53:49:15,649	0,151980	1350
52	30	45:38:44,406	0,152191	1140
53	30	55:14:19,541	0,151658	1440
54	30	53:24:47,513	0,151901	1320
55	30	119:38:25,428	0,155957	3030
56	30	73:32:51,730	0,152807	1560
57	30	147:06:03,005	0,157046	3030
58	30	64:22:33,793	0,153970	1350
59	30	63:05:47,246	0,152364	1200
60	30	71:50:29,155	0,152012	1380
61	30	162:12:22,761	0,156653	3030
62	35	179:28:38,652	0,156299	3535
63	35	64:40:20,415	0,150038	1225
64	35	79:27:15,414	0,153467	1505
65	35	70:33:55,142	0,153702	1365
66	35	193:57:05,561	0,156903	3535
67	35	91:58:01,522	0,145221	1365
68	35	127:17:59,626	0,152408	1855
69	35	107:16:23,186	0,15564	1365
70	35	212:20:45:54,963	0,156603	3535

**Tabelle C.2:** PSO - Populationsgröße:  
Ergebnisse der Einzelläufe - Modell 2

Nr	iiw	fiw	n	Laufzeit (hh:mm:ss,000)	maxMSE	Anzahl Simulationen
1	0,9	0,2	10	22:02:27,890	0,750278	2020
2	0,9	0,2	10	21:11:27,792	0,751186	2020
3	0,9	0,2	10	23:17:45,109	0,747172	2020
4	0,9	0,2	10	21:25:20,984	0,746895	2020
5	0,9	0,2	10	21:15:20,396	0,74608	2020
6	0,9	0,4	10	21:45:56,879	0,746429	2020
7	0,9	0,4	10	22:01:51,878	0,744089	2020
8	0,9	0,4	10	22:04:33,945	0,74396	2020
9	0,9	0,4	10	22:01:33,510	0,744497	2020
10	0,9	0,4	10	24:14:26,710	0,747516	2020
11	0,9	0,2	20	20:54:12,660	0,749334	2020
12	0,9	0,2	20	27:53:33,710	0,750501	2020
13	0,9	0,2	20	22:12:14,139	0,74969	2020
14	0,9	0,2	20	20:54:11,013	0,747491	2020
15	0,9	0,2	20	24:09:08,038	0,746723	2020
16	0,9	0,2	20	21:19:47,393	0,746851	2020
17	0,9	0,2	20	22:16:58,941	0,749293	2020
18	0,9	0,2	20	22:14:14,388	0,747435	2020
19	0,9	0,2	20	21:46:24,463	0,74595	2020
20	0,9	0,2	20	23:10:56,321	0,747208	2020
21	0,9	0,4	20	20:57:28,587	0,752861	2020
22	0,9	0,4	20	21:05:13,813	0,742965	2020
23	0,9	0,4	20	21:01:37,398	0,74744	2020
24	0,9	0,4	20	21:07:30,318	0,750113	2020
25	0,9	0,4	20	21:05:39,548	0,748046	2020
26	0,9	0,4	20	21:06:38,847	0,745356	2020

27	0,9	0,4	20	21:04:52,619	0,749034	2020
28	0,9	0,4	20	20:50:13,359	0,748995	2020
29	0,9	0,4	20	22:8:38,796	0,742711	2020
30	0,9	0,4	20	28:01:09,836	0,749752	2020
31	0,9	0,4	30	23:02:32,618	0,749915	2020
32	0,9	0,4	30	21:12:41,563	0,746137	2020
33	0,9	0,4	30	20:52:25,433	0,746403	2020
34	0,9	0,4	30	20:58:02,660	0,744457	2020
35	0,9	0,4	30	20:41:17,134	0,745775	2020
36	0,9	0,4	30	22:09:16,299	0,743347	2020
37	0,9	0,4	30	07:59:02,928	0,738606	740
38	0,9	0,4	30	23:03:04,199	0,746163	2020
39	0,9	0,4	30	11:04:00,805	0,737702	1060
40	0,9	0,4	30	21:02:17,450	0,746258	2020
41	0,9	0,4	40	21:33:51,603	0,748978	2020
42	0,9	0,4	40	20:53:29,606	0,743227	2020
43	0,9	0,4	40	21:03:30,601	0,750466	2020
44	0,9	0,4	40	20:59:36,087	0,749145	2020
45	0,9	0,4	40	22:45:18,499	0,750651	2020
46	0,9	0,4	40	09:44:51,009	0,729503	940
47	0,9	0,4	40	07:51:22,059	0,724235	760
48	0,9	0,4	40	21:08:15,602	0,744105	2020
49	0,9	0,4	40	20:43:31,747	0,74932	2020
50	0,9	0,4	40	22:13:48,724	0,748717	2020
51	0,9	0,4	50	21:29:09,403	0,744222	2020
52	0,9	0,4	50	08:00:36,715	0,734408	780
53	0,9	0,4	50	20:51:51,742	0,745558	2020
54	0,9	0,4	50	21:05:31,565	0,751679	2020
55	0,9	0,4	50	22:50:57,867	0,749209	2020
56	0,9	0,4	60	10:55:39,302	0,722476	1000
57	0,9	0,4	60	10:38:26,931	0,737128	1040
58	0,9	0,4	60	11:05:36,915	0,720051	1080
59	0,9	0,4	60	10:54:30,214	0,730263	1060
60	0,9	0,4	60	20:59:50,166	0,740857	2020
61	0,9	0,4	70	20:54:32,470	0,74249	2020
62	0,9	0,4	70	22:36:53,111	0,740035	2020
63	0,9	0,4	70	12:36:19,434	0,735146	1160
64	0,9	0,4	70	11:01:50,600	0,72111	1060
65	0,9	0,4	70	20:51:21,136	0,746015	2020

**Tabelle C.3:** PSO - Trägheitsgewichtung:  
Ergebnisse der Einzelläufe - Modell 1

Nr	iiw	fiw	n	Laufzeit (hh:mm:ss,000)	maxMSE	Anzahl Simulationen
1	0,9	0,4	10	82:47:36,503	0,156163	2020
2	0,9	0,4	10	82:58:15,978	0,158068	2020
3	0,9	0,4	10	82:43:40,918	0,155279	2020
4	0,9	0,4	10	83:41:41,413	0,157383	2020
5	0,9	0,4	10	83:41:40,354	0,157521	2020
6	0,9	0,2	20	81:46:50,634	0,156697	2020
7	0,9	0,2	20	79:45:43,911	0,157800	2020
8	0,9	0,2	20	80:06:29,076	0,157804	2020
9	0,9	0,2	20	80:00:02,538	0,156268	2020
10	0,9	0,2	20	79:54:57,395	0,155999	2020
11	0,9	0,4	20	83:51:19,400	0,157372	2020
12	0,9	0,4	20	28:01:40,384	0,154501	680
13	0,9	0,4	20	83:57:54,376	0,157227	2020
14	0,9	0,4	20	83:37:35,879	0,157457	2020
15	0,9	0,4	20	83:39:08,697	0,157159	2020
16	0,9	0,4	30	80:13:18,636	0,156149	2020
17	0,9	0,4	30	80:08:12,832	0,157025	2020
18	0,9	0,4	30	74:48:14,403	0,155426	1880
19	0,9	0,4	30	80:09:50,381	0,156724	2020
20	0,9	0,4	30	63:06:02,752	0,156017	1580
21	0,9	0,4	40	81:55:07,187	0,157001	2020
22	0,9	0,4	40	81:54:41,093	0,156473	2020
23	0,9	0,4	40	30:48:35,681	0,152964	720
24	0,9	0,4	40	34:46:50,678	0,153102	820
25	0,9	0,4	40	39:27:09,749	0,154424	940

**Tabelle C.4:** PSO - Trägheitsgewichtung:  
Ergebnisse der Einzelläufe - Modell 2

Nr	acc1	acc2	Laufzeit (hh:mm:ss,000)	maxMSE	Anzahl Simulationen
1	2	1,8	23:25:38,922	0,744883	2020
2	2	1,8	23:34:19,831	0,747008	2020
3	2	1,8	21:34:27,959	0,747464	2020
4	2	1,8	21:49:47,706	0,748494	2020
5	2	1,8	18:36:57,626	0,748036	1800
6	2	1,8	21:31:13,136	0,748212	2020
7	2	1,8	21:41:16,654	0,74756	2020
8	2	1,8	19:42:42,420	0,748319	1820
9	2	1,8	21:45:56,384	0,747424	2020
10	2	1,8	21:50:52,268	0,746015	2020
11	2	2,2	21:40:49,551	0,74565	1880
12	2	2,2	23:21:12,812	0,745	2020
13	2	2,2	21:39:10,809	0,746636	2020
14	2	2,2	21:37:42,301	0,75229	2020
15	2	2,2	20:45:06,989	0,752125	2020
16	2	2,2	24:08:33,881	0,745999	2020
17	2	2,2	24:07:20,939	0,746536	2020
18	2	2,2	22:05:50,550	0,746926	1860
19	2	2,2	24:02:10,779	0,751182	2020
20	2	2,2	24:10:49,873	0,745241	2020
21	2	2,2	24:02:05,911	0,747312	2020
22	2	2,2	18:38:39,684	0,749589	1540
23	2	2,2	24:02:48,865	0,751854	2020
24	2	2,4	24:09:16,385	0,748503	2020
25	2	2,4	17:05:05,950	0,749023	1640
26	2	2,4	24:08:23,088	0,746429	2020
27	2	2,4	20:51:34,415	0,750519	2020
28	2	2,4	21:22:18,522	0,750378	2020
29	2	2,4	21:18:56,894	0,747605	2020
30	2	2,4	21:22:49,084	0,745055	2020
31	2	2,4	21:25:19,180	0,75088	2020
32	2	2,4	20:59:44,496	0,747316	2020
33	2	2,4	21:02:50,003	0,74426	2020
34	2	2,4	118:56:23,712	0,746782	1820
35	2	2,4	20:58:42,603	0,75327	2020
36	1,8	2	21:20:26,737	0,744749	2020
37	1,8	2	21:23:08,463	0,747022	2020
38	1,8	2	21:01:22,085	0,748805	2020
39	1,8	2	21:15:06,679	0,746953	2020
40	1,8	2	21:15:32,383	0,747215	2020
41	2	1	16:05:01,003	0,735216	1540
42	2	1	20:10:48,522	0,735283	1940
43	2	1	18:45:07,842	0,739618	1760
44	2	3	20:57:06,814	0,745471	2020
45	2	3	20:57:16,049	0,749892	2020
46	2	3	18:07:59,092	0,749257	1740
47	2	3	20:54:31,605	0,747971	2020
48	2	3	20:46:48,634	0,744381	2020
49	2,2	2	21:02:11,945	0,745178	2020
50	2,2	2	21:04:20,389	0,750802	2020
51	2,2	2	20:59:39,393	0,7449	2020
52	2,2	2	21:01:42,257	0,748364	2020
53	2,2	2	21:00:51,287	0,751663	2020

**Tabelle C.5:** PSO - Beschleunigungskoeffizienten:  
Ergebnisse der Einzelläufe - Modell 1

Nr	Methode	Laufzeit (hh:mm:ss,000)	maxMSE	Anzahl Simulationen
1	saturation at limit	22:16:18,077	0,751982	1840
2	saturation at limit	16:45:35,862	0,745825	1320
3	saturation at limit	16:04:16,036	0,751983	1340
4	saturation at limit	22:59:01,745	0,750714	1920
5	saturation at limit	19:27:34,116	0,750264	1600
6	saturation at limit	22:35:10,252	0,749892	1520
7	saturation at limit	20:43:23,139	0,747265	1360
8	saturation at limit	18:51:10,291	0,749904	1200
9	saturation at limit	16:08:46,201	0,746008	1040
10	saturation at limit	24:56:17,753	0,749500	1720
11	bounce-off	20:54:12,660	0,749334	2020
12	bounce-off	27:53:33,710	0,750501	2020
13	bounce-off	22:12:14,139	0,749690	2020
14	bounce-off	20:54:11,013	0,747491	2020
15	bounce-off	24:09:08,038	0,746723	2020
16	bounce-off	21:19:47,393	0,746851	2020
17	bounce-off	22:16:58,941	0,749293	2020
18	bounce-off	22:14:14,388	0,747435	2020
19	bounce-off	21:46:24,463	0,745950	2020
20	bounce-off	23:10:56,321	0,747208	2020

**Tabelle C.6:** PSO - Umgang mit Wertebereichsgrenzen:  
Ergebnisse der Einzelläufe - Modell 1

Nr	Methode	Laufzeit (hh:mm:ss,000)	maxMSE	Anzahl Simulationen
1	saturation at limit	66:02:25,000	0,155239	1640
2	saturation at limit	58:16:40,078	0,157952	1440
3	saturation at limit	68:04:01,564	0,157345	1680
4	saturation at limit	81:28:15,433	0,158562	2020
5	saturation at limit	82:58:27,528	0,158299	2020
6	saturation at limit	82:15:35,508	0,155558	2020
7	saturation at limit	80:17:16,265	0,159005	2020
8	saturation at limit	81:41:33,832	0,15558	2020
9	saturation at limit	82:02:39,827	0,157881	2020
10	saturation at limit	75:47:59,748	0,158249	1880
11	saturation at limit	70:17:49,537	0,15754	1740

**Tabelle C.7:** PSO - Umgang mit Wertebereichsgrenzen:  
Ergebnisse der Einzelläufe - Modell 2

# D Weitere Ergebnisse der Signalintegritätsanalyse

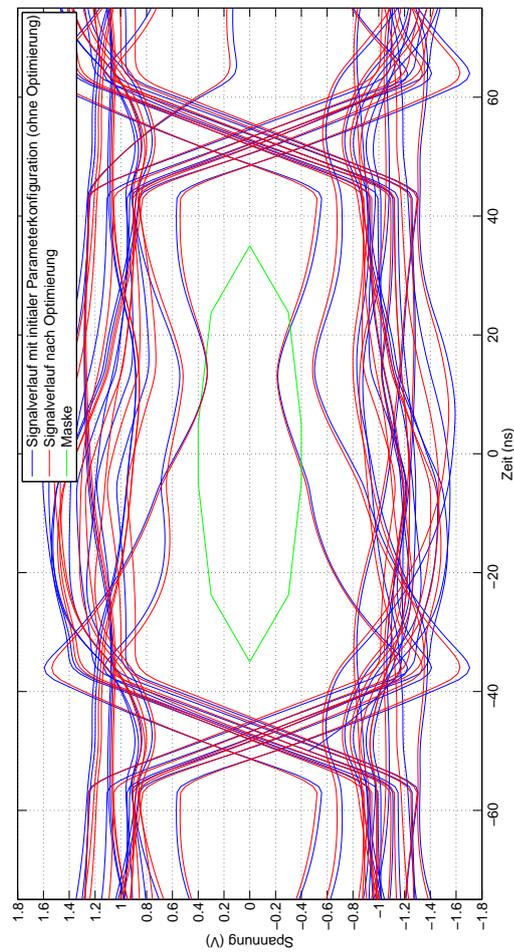


Abbildung D.1: Augendiagramm zu den Optimierungen mittels SA für ECU 1

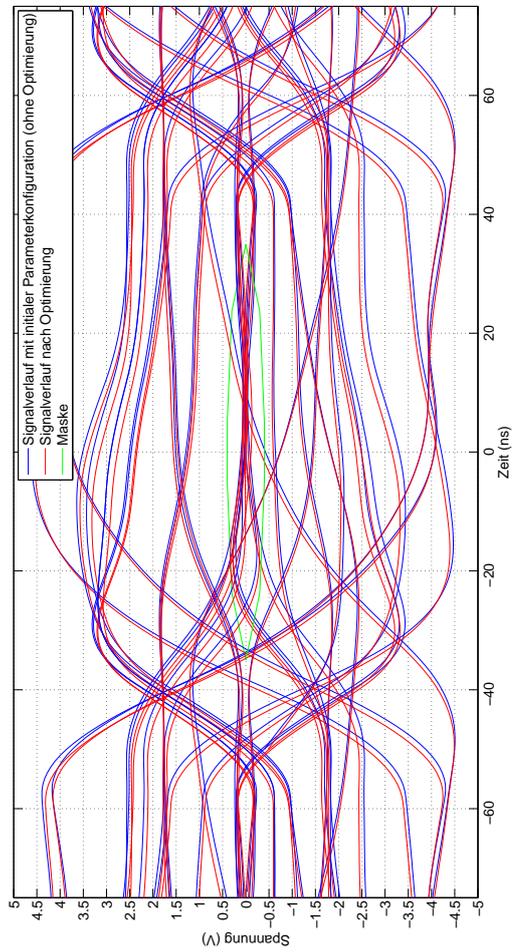


Abbildung D.2: Augendiagramm zu den Optimierungen mittels SA für ECU 2

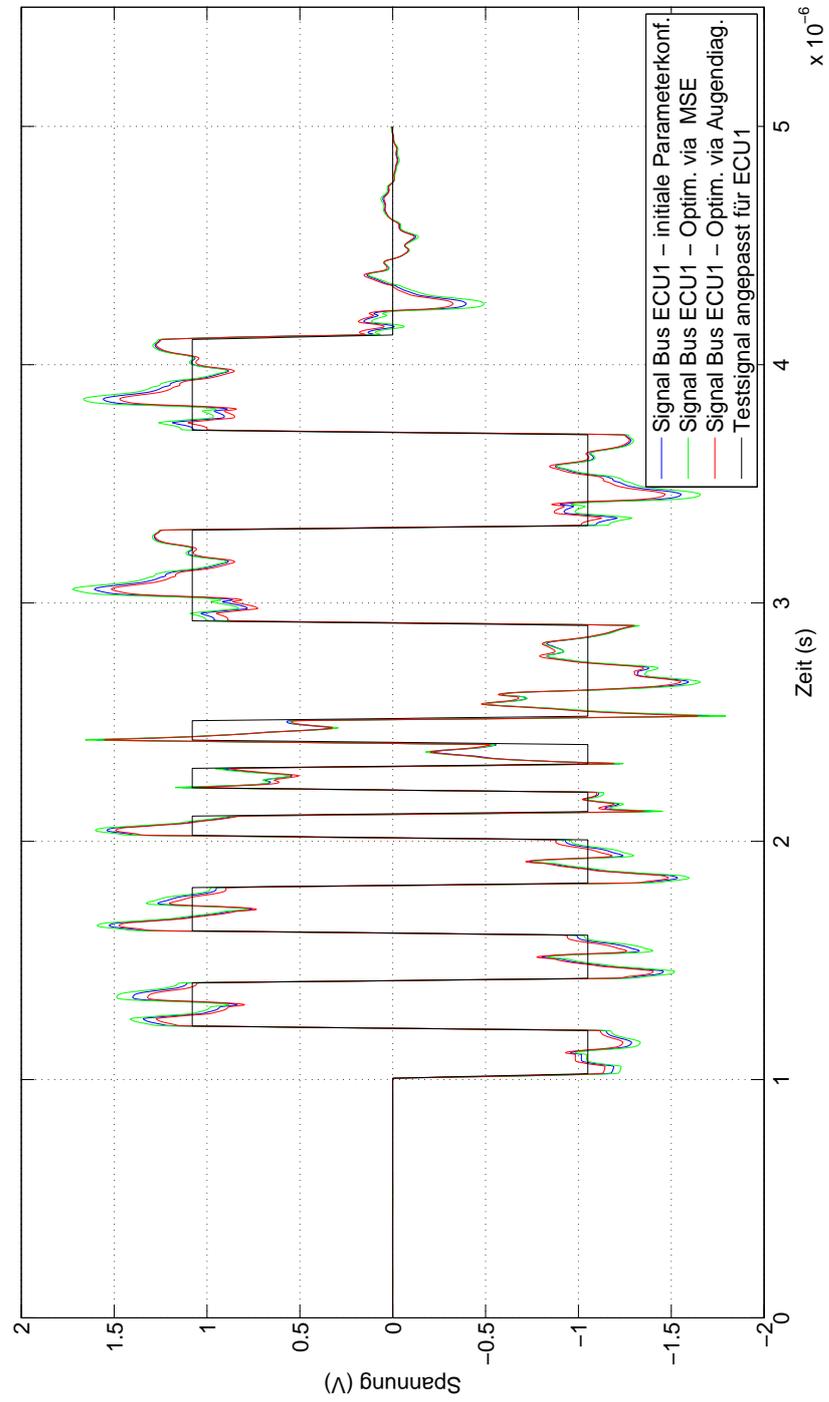


Abbildung D.3: Vergleich der Signalverläufe an ECU 1

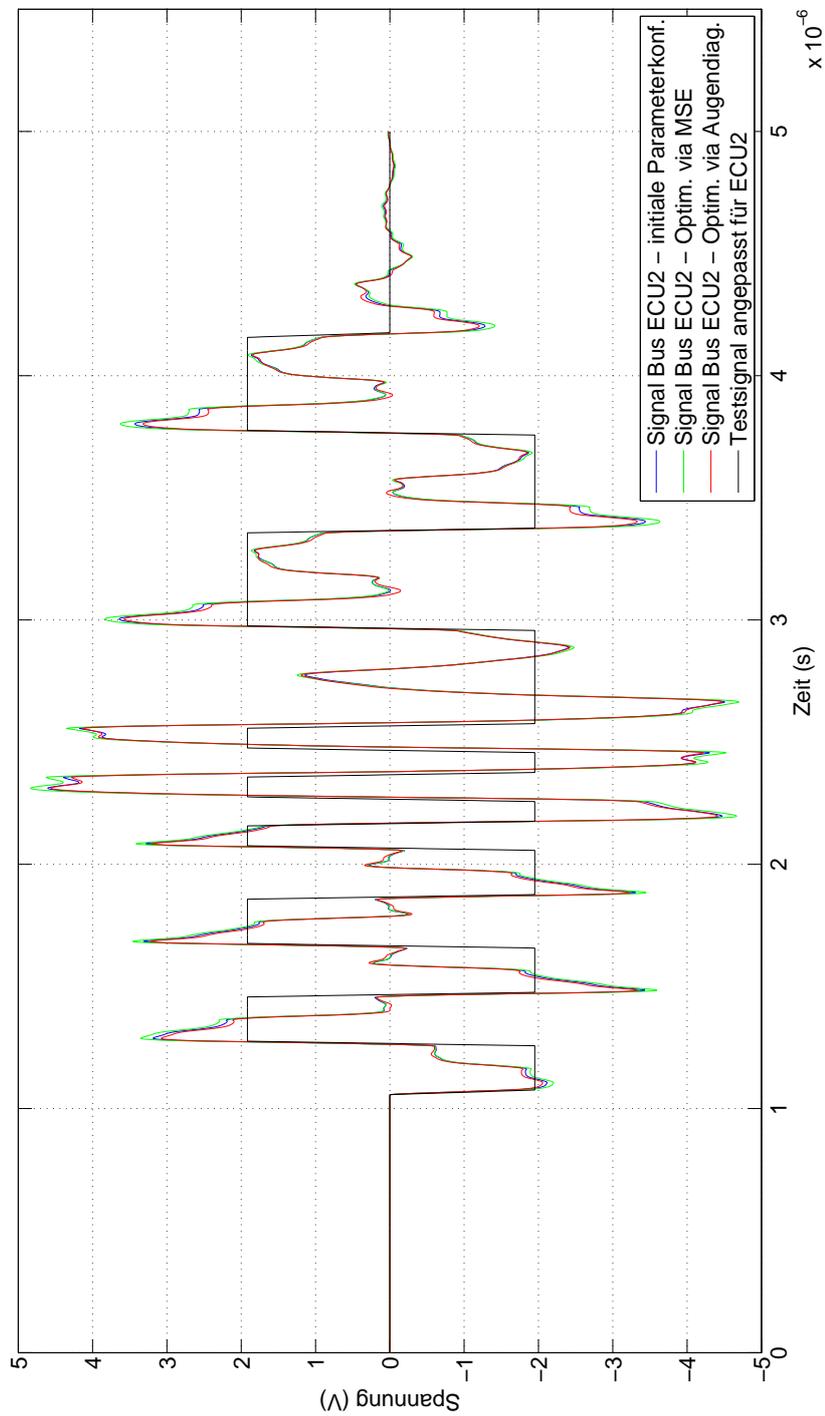


Abbildung D.4: Vergleich der Signalverläufe an ECU 2

# Abbildungsverzeichnis

2.1	Komponenten des FlexRay Physical Layer . . . . .	9
2.2	Signalpegel FlexRay-Bussystem . . . . .	9
2.3	Punkt-zu-Punkt-Verbindung zweier Knoten . . . . .	11
2.4	Verbindung von vier Knoten über einen passiven Bus . . . . .	12
2.5	Verbindung zwischen vier Knoten mittels passivem Stern . . . . .	12
2.6	Analyseumgebung [Gü07] . . . . .	13
3.1	Hill-Climbing . . . . .	21
3.2	Tabu-Suche . . . . .	22
3.3	schematischer Ablauf Tabu-Suche nach [GKK04] . . . . .	23
3.4	Simulated Annealing im Vergleich zu Hill-Climbing . . . . .	25
3.5	Kontrollfluss eines genetischen Algorithmus nach [BK05] . . . . .	33
3.6	stochastisch uniforme Selektion . . . . .	35
3.7	Roulette-Selektion . . . . .	35
3.8	binärer Crossover . . . . .	36
3.9	1-Punkt Crossover . . . . .	37
3.10	2-Punkt Crossover . . . . .	38
3.11	Beispiel eines intermediären Crossover mit drei Parametern . . . . .	39
3.12	Probleme bei proportionaler Fitness-Skalierung . . . . .	42
3.13	Schematischer Ablauf PSO nach [BK05] . . . . .	44
4.1	Topologie von Modell 1 . . . . .	50
4.2	Topologie von Modell 2 . . . . .	52
4.3	SA - Konvergenzverhalten - Modell 1 . . . . .	57
4.4	SA - Konvergenzverhalten - Modell 2 . . . . .	59
4.5	SA - Lokalisierungsschritte für $\alpha = 0,7$ : Konvergenzverhalten - Modell 1 . . . . .	61
4.6	SA - Lokalisierungsschritte für $\alpha = 0,7$ : Konvergenzverhalten - Modell 2 . . . . .	63
4.7	GA - Populationsgröße: Konvergenzverhalten - Modell 1 . . . . .	66
4.8	GA - Populationsgröße: Konvergenzverhalten - Modell 2 . . . . .	68
4.9	GA - Turnierselektion: Konvergenzverhalten - Modell 1 . . . . .	73

4.10	GA - Konvergenzverläufe unterschiedlicher Verhältniswerte zwischen Crossover und Mutation - Modell 1 . . . . .	77
4.11	PSO - Populationsgröße: Konvergenzverhalten - Modell 1 . . . . .	81
4.12	PSO - Populationsgröße: Konvergenzverhalten - Modell 2 . . . . .	82
4.13	PSO - Trägheitsgewicht - Modell 1 . . . . .	86
4.14	PSO - Trägheitsgewicht - Modell 2 . . . . .	87
4.15	PSO - Umgang mit Parameterwertebereichsgrenzen: Konvergenzverhalten - Modell 1 . . . . .	89
4.16	PSO - Umgang mit Parameterwertebereichsgrenzen: Konvergenzverhalten - Modell 2 . . . . .	91
4.17	Vergleich der Konvergenzverläufe des Simulated Annealing und der Partikel Schwarm Optimierung nach Anpassung der Algorithmen für Modell 1 . . . .	95
4.18	Vergleich der Konvergenzverläufe des Simulated Annealing und der Partikel Schwarm Optimierung nach Anpassung der Algorithmen für Modell 2 . . . .	95
5.1	Topologie des Testmodells . . . . .	98
5.2	Maske zur Bestimmung der Signalintegrität mittels Augendiagramm . . . . .	99
5.3	Ausschnitt aus den Signalverläufen an ECU 1 . . . . .	102
5.4	Ausschnitt aus den Signalverläufen an ECU 2 . . . . .	103
A.1	SA - Lokalisierungsschritte für $\alpha = 0,8$ : Konvergenzverhalten - Modell 1 . . .	113
A.2	SA - Lokalisierungsschritte für $\alpha = 0,9$ : Konvergenzverhalten - Modell 1 . . .	114
A.3	SA - Lokalisierungsschritte für $\alpha = 0,8$ : Konvergenzverhalten - Modell 2 . . .	115
A.4	SA - Lokalisierungsschritte für $\alpha = 0,9$ : Konvergenzverhalten - Modell 2 . . .	116
B.1	GA - Rest-Selektion: Konvergenzverhalten - Modell 1 . . . . .	120
B.2	GA - Roulette-Selektion: Konvergenzverhalten - Modell 1 . . . . .	122
B.3	GA - stochastisch uniforme Selektion: Konvergenzverhalten - Modell 1 . . . .	122
D.1	Augendiagramm zu den Optimierungen mittels SA für ECU 1 . . . . .	133
D.2	Augendiagramm zu den Optimierungen mittels SA für ECU 2 . . . . .	134
D.3	Vergleich der Signalverläufe an ECU 1 . . . . .	135
D.4	Vergleich der Signalverläufe an ECU 2 . . . . .	136

# Tabellenverzeichnis

2.1	Elemente zum Aufbau eines FlexRay Systems . . . . .	11
3.1	Terminologie der genetischen Algorithmen, nach [Nis97] . . . . .	31
4.1	Modell 1 - Fehlerwerte mit initialen Parameterwerten . . . . .	51
4.2	Modell 2 - Fehlerwerte mit initialen Parameterwerten . . . . .	52
4.3	SA - initiale Analyse - Modell 1 . . . . .	54
4.4	SA - initiale Analyse - Modell 2 . . . . .	55
4.5	SA - Einfluss des Parameters $\alpha$ - Modell 1 . . . . .	57
4.6	SA - Einfluss des Parameters $\alpha$ - Modell 2 . . . . .	58
4.7	SA - Einfluss Lokalisierungsschritte - Modell 1 . . . . .	60
4.8	SA - Einfluss Lokalisierungsschritte - Modell 2 . . . . .	62
4.9	GA - Populationsgröße - Modell 1 . . . . .	67
4.10	GA - Populationsgröße - Modell 2 . . . . .	68
4.11	GA - stochastisch uniforme Selektion - Modell 1 . . . . .	70
4.12	GA - intermediärer Crossover: Einfluss des Ratio-Parameters - Modell 1 . . . . .	70
4.13	GA - Selektions- und Crossover-Verfahren - Modell 1 . . . . .	72
4.14	GA - Selektions- und Crossover-Verfahren - Modell 2 . . . . .	74
4.15	GA - Verhältnis zwischen Crossover und Mutation - Modell 1 . . . . .	77
4.16	Initiale Werte der Ablaufparameter des PSO . . . . .	79
4.17	PSO - Populationsgröße - Modell 1 . . . . .	81
4.18	PSO - Populationsgröße - Modell 2 . . . . .	82
4.19	PSO - Trägheitsgewicht - Modell 1 . . . . .	85
4.20	PSO - Trägheitsgewicht: $fiw = 0,2$ - Modell 1 . . . . .	85
4.21	PSO - Trägheitsgewicht - Modell 2 . . . . .	86
4.22	PSO - Beschleunigungskoeffizienten - Modell 1 . . . . .	88
4.23	PSO - Umgang mit Wertebereichsgrenzen - Modell 1 . . . . .	89
4.24	PSO - Umgang mit Wertebereichsgrenzen - Modell 2 . . . . .	90
4.25	Auswertung der Optimierungsergebnisse der Algorithmen - Modell 1 . . . . .	92
4.26	Auswertung der Optimierungsergebnisse der Algorithmen - Modell 2 . . . . .	93

5.1	Vergleich der ermittelten Fehlerwerte aus den Busanalysen . . . . .	100
5.2	Vergleich der ermittelten Busparameterkonfigurationen mit dem größten Fehlerwert . . . . .	101
A.1	SA - initiale Analyse: Ergebnisse der Einzelläufe - Modell 1 . . . . .	107
A.2	SA - initiale Analyse: Ergebnisse der Einzelläufe - Modell 2 . . . . .	108
A.3	SA - Parameter $\alpha = 0,7$ : Ergebnisse der Einzelläufe - Modell 1 . . . . .	108
A.4	SA - Parameter $\alpha = 0,9$ : Ergebnisse der Einzelläufe - Modell 1 . . . . .	108
A.5	SA - Parameter $\alpha = 0,7$ : Ergebnisse der Einzelläufe - Modell 2 . . . . .	109
A.6	SA - Parameter $\alpha = 0,9$ : Ergebnisse der Einzelläufe - Modell 2 . . . . .	109
A.7	SA - Lokalisierungsschritte, mit Schrittweite 2,5 %: Ergebnisse der Einzelläufe - Modell 1 . . . . .	110
A.8	SA - Lokalisierungsschritte, mit Schrittweite 5 %: Ergebnisse der Einzelläufe - Modell 1 . . . . .	111
A.9	SA - Lokalisierungsschritte, mit Schrittweite 10 %: Ergebnisse der Einzelläufe - Modell 1 . . . . .	112
A.10	SA - Lokalisierungsschritte, mit Schrittweite 2,5 %: Ergebnisse der Einzelläufe - Modell 2 . . . . .	113
A.11	SA - Lokalisierungsschritte, mit Schrittweite 5 %: Ergebnisse der Einzelläufe - Modell 2 . . . . .	114
A.12	SA - Lokalisierungsschritte, mit Schrittweite 10 %: Ergebnisse der Einzelläufe - Modell 2 . . . . .	115
B.1	GA - Populationsgröße: Ergebnisse der Einzelläufe - Modell 1 . . . . .	118
B.2	GA - Populationsgröße: Ergebnisse der Einzelläufe - Modell 2 . . . . .	120
B.3	GA - Crossover-Operatoren bei stochastisch uniformer Selektion - Modell 1 .	121
B.4	GA - Verhältnis zwischen Crossover und Mutation: Einzelergebnisse - Modell 1	123
C.1	PSO - Populationsgröße: Ergebnisse der Einzelläufe - Modell 1 . . . . .	126
C.2	PSO - Populationsgröße: Ergebnisse der Einzelläufe - Modell 2 . . . . .	128
C.3	PSO - Trägheitsgewichtung: Ergebnisse der Einzelläufe - Modell 1 . . . . .	129
C.4	PSO - Trägheitsgewichtung: Ergebnisse der Einzelläufe - Modell 2 . . . . .	130
C.5	PSO - Beschleunigungskoeffizienten: Ergebnisse der Einzelläufe - Modell 1 .	131
C.6	PSO - Umgang mit Wertebereichsgrenzen: Ergebnisse der Einzelläufe - Modell 1 . . . . .	132
C.7	PSO - Umgang mit Wertebereichsgrenzen: Ergebnisse der Einzelläufe - Modell 2 . . . . .	132

# Literaturverzeichnis

- [Bak85] BAKER, J. E.: Adaptive selection methods for genetic algorithms. In: *Proc. Int. Conf. on Genetic Algorithms and Their Applications*, 1985, S. 101–111
- [BK05] BURKE, Edmund K. ; KENDALL, Graham: *Search Methodologies - Introductory Tutorials in Optimization and Decision Support Techniques*. Springer, 2005
- [Bri03] BRIGE, Brian: TPSoT-A Particle Swarm Optimization Toolbox for use with Matlab. In: *Proceedings of the 2003 IEEE Swarm Intelligence Symposium (2003)*, S. 182–186
- [Dar63] DARWIN, Charles: *On the origin of species by means of natural selection or the preservation of favoured races in the struggle for life*. 5. print. New York : New American Libr., 1963
- [DMC96] DORIGO, M. ; MANIEZZO, V. ; COLORNI, A.: The ant system: Optimization by a colony of cooperating agents. In: *IEEE Trans. on Systems, Man, and Cybernetics - Part B* 26 (1996), Nr. 1, S. 29–41
- [DO05] DIAZ ORTEGA, Lorena: *Physical Layer Modellierung der Bussysteme CAN und FlexRay im Kraftfahrzeug*. Aachen : Shaker, 2005
- [Dor05] DORF, Richard C. (Hrsg.): *Systems, Controls, Embedded Systems, Energy, and Machines - Electrical Engineering Handbook*. Third Edition. Bonn : Taylor & Francis, Inc., 2005
- [Fle06a] FlexRay Consortium: *FlexRay Electrical Physical Layer Application Notes*. Version 2.1 Revision B. November 2006. – Erhältlich unter <http://www.flexray.com>
- [Fle06b] FlexRay Consortium: *FlexRay Electrical Physical Layer Specification*. Version 2.1 Revision B. November 2006. – Erhältlich unter <http://www.flexray.com>
- [GBF<sup>+</sup>02] GIUSTO, Paolo ; BRUNEL, Jean-Yves ; FERRARI, Alberto ; FOURGEAU, Eliane ; LAVAGNO, Luciano ; SANGIOVANNI-VINCENTELLI, Alberto: Automotive Vir-

- tual Integration Platforms: Why's, What's, and How's. In: *Proceedings of the 2002 IEEE International Conference on Computer Design 0* (2002)
- [GKK04] GERDES, Ingrid ; KLAWONN, Frank ; KRUSE, Rudolf: *Evolutionäre Algorithmen*. Vieweg, 2004
- [Glo86] GLOVER, Fred: Future paths for integer programming and links to artificial intelligence. In: *Comput. Oper. Res.* 13 (1986), Nr. 5, S. 533–549. – ISSN 0305–0548
- [Glo89] GLOVER, Fred: Tabu search - part i. In: *OSRA Journal on Computing* 1 (1989), Nr. 3, S. 190–206
- [Glo90] GLOVER, Fred: Tabu search - part ii. In: *OSRA Journal on Computing* 2 (1990), Nr. 1, S. 4–32
- [Grz08] GRZEMBA, Andreas (Hrsg.): *MOST - The Automotive Multimedia Network*. Poing : Franzis, 2008
- [Gü07] GÜNTHER, Harald: *Simulationsbasierte Bewertung der Übertragungssicherheit von Kfz-Bussystemen unter Berücksichtigung von Parameterstreuungen*, Universität Dortmund, Diplomarbeit, 2007
- [HKL07] HIETL, Hubert ; KÖTZ, Jens ; LINN, Günter: Bereit für FlexRay - FlexRay-Serieneinführung bei Audi. In: *Elektronik automotive* 1 (2007), 32–36. <http://www.elektroniknet.de/home/automotive/technik-know-how/uebersicht/1/bussysteme/bereit-fuer-flexray/>, Abruf: 27.07.2009
- [Hol92] HOLLAND, John H.: *Adaptation in natural and artificial systems*. Nachdruck. Cambridge, MA, USA : MIT Press, 1992
- [Kar05] KARRENBERG, Ulrich: *Signale - Prozesse - Systeme: Eine multimediale und interaktive Einführung in die Signalverarbeitung*. 4., neu bearb. u. erw. Aufl. Berlin : Springer, 2005
- [KE95] KENNEDY, J. ; EBERHART, R.: Particle swarm optimization. In: *Proceedings of the 1995 IEEE International Conference on Neural Networks*, 1995, S. 1942–1948
- [Kin94] KINNEBROCK, Werner: *Optimierung mit genetischen und selektiven Algorithmen*. Oldenbourg, 1994
- [Lip05] LIPPE, Wolfram-Manfred: *Soft-Computing mit Neuronalen Netzen, Fuzzy-Logic und Evolutionären Algorithmen*. Berlin : Springer, 2005

- [Liu09] LIU, Yang: *Evaluierung und Umsetzung von Parallelisierungsansätzen zur effektiven Untersuchung der Übertragungssicherheit von Kfz-Bussystemen*, Technische Universität Dortmund, Diplomarbeit, 2009
- [mat08] The MathWorks: *Matlab R2008b, Genetic Algorithm and Direct Search Toolbox 2.4.2*. <http://www.mathworks.com/products/gads/>. Version: 2008, Abruf: 27.07.2009. – Dokumentation online verfügbar
- [Mit96] MITCHELL, M.: *An Introduction to Genetic Algorithms*. Cambridge, MA, USA : MIT Press, 1996
- [MRR<sup>+</sup>53] METROPOLIS, N. ; ROSENBLUTH, A. W. ; ROSENBLUTH, M. N. ; TELLER, A. H. ; TELLER, E.: Equation of State Calculations by Fast Computing Machines. In: *Journal of Chemical Physics* 21 (1953), June, S. 1087–1092
- [Nis97] NISSEN, Volker: *Einführung in Evolutionäre Algorithmen - Optimierung nach dem Vorbild der Evolution*. Wiesbaden : Vieweg, 1997
- [Nus09] NUSZKOWSKI, Heinrich: *Digitale Signalübertragung: Grundlagen der digitalen Nachrichtenübertragungssysteme*. 2., neu bearb. u. erg. Aufl. Dresden : Vogt, 2009
- [OBP02] OURIQUE, C.O. ; BISCAIA, E.C. ; PINTO, J.C.: The use of particle swarm optimization for dynamical analysis in chemical processes. In: *Computers and Chemical Engineering* 26 (2002), Nr. 12, S. 4–32
- [Poh98] POHLHEIM, Hartmut: *Entwicklung und systemtechnische Anwendung Evolutionärer Algorithmen*. Ilmenau, Deutschland, Technische Universität Ilmenau, Dissertation, 1998
- [Rau08] RAUSCH, Mathias (Hrsg.): *FlexRay - Grundlagen, Funktionsweise, Anwendung*. München : Hanser, 2008
- [Rud93] RUDOLPH, Günther: Massively parallel simulated annealing and its relation to evolutionary algorithms. In: *Bericht / Systems Analysis Research Group ; 93,1* (1993), S. 134–139
- [SE98a] SHI, Yuhui ; EBERHART, Russell C.: A modified particle swarm optimizer. In: *Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on*, 1998, S. 69–73
- [SE98b] SHI, Yuhui ; EBERHART, Russell C.: Parameter Selection in Particle Swarm Optimization. In: *EP '98: Proceedings of the 7th International Conference on*

- Evolutionary Programming VII*. London, UK : Springer-Verlag, 1998, S. 591–600
- [SHF94] SCHÖNEBURG, Eberhard ; HEINZMANN, Frank ; FEDDERSEN, Sven: *Genetische Algorithmen und Evolutionsstrategien - Eine Einführung in Theorie und Praxis der simulierten Evolution*. Bonn : Addison-Wesley, 1994
- [Vöc08] VÖCKING, Berthold (Hrsg.): *Taschenbuch der Algorithmen*. Dritte Auflage. Berlin : Springer, 2008
- [Weg03] WEGENER, Ingo: *Komplexitätstheorie*. Berlin : Springer, 2003
- [WH] WEICH, Carsten ; HOFMANN, Roman: *Oben angekommen - Der Einsatz von FlexRay in einem Oberklasse-Fahrzeug*. <http://www.elektroniknet.de/home/automotive/technik-know-how/uebersicht/1/bussysteme/oben-angekommen/>, Abruf: 27.07.2009
- [WHW09] *Kapitel* Das mechatronische Fahrwerk der Zukunft. In: WINNER, Hermann (Hrsg.) ; HAKULI, Stephan (Hrsg.) ; WOLF, Gabriele (Hrsg.): *Handbuch Fahrerassistenzsysteme - Grundlagen, Komponenten und Systeme für aktive Sicherheit und Komfort*. Wiesbaden : Vieweg+Teubner, 2009
- [ZS08] ZIMMERMANN, Werner ; SCHMIDGALL, Ralf: *Bussysteme in der Fahrzeugtechnik - Protokolle und Standards*. Dritte Auflage. Wiesbaden : Vieweg, 2008
- [ZYH03] ZAHNG, Liping ; YU, Huanjun ; HU, Shangxu: A new approach to improve particle swarm optimization. In: *Genetic and evolutionary computation - GECCO 2003* (2003), S. 134–139

Hiermit versichere ich, dass ich die vorliegende Arbeit selbstständig verfasst habe und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet sowie Zitate kenntlich gemacht habe.

Dortmund, den 8. Oktober 2009

Manuel Beer

