

7 Der Genetische Algorithmus

Dieses Kapitel beschäftigt sich mit Fragestellungen zu dem im Rahmen dieser Arbeit eingesetzten Genetischen Algorithmus selbst. Für seine Verwendung in einem automatisierten Auswahlverfahren war es, aufbauend auf die in 4.3 vorgestellten Grundfunktionen eines GA notwendig, Techniken und Verfahren zu entwerfen, die eine möglichst hohe Robustheit und Konvergenz garantieren. Zusätzlich sollte die benötigte Rechenzeit, die ein GA für eine Optimierung benötigt, so kurz wie möglich gehalten werden.

Wie in Abschnitt 4.3.2 bereits angesprochen, stellen die Konvergenz und Robustheit eines Genetischen Algorithmus (GA) wesentliche Punkte in bezug auf die Beurteilung seiner Leistungsfähigkeit dar.

Im Gegensatz zu deterministischen und heuristischen Verfahren besitzen stochastische Systeme wie GAs nicht die Eigenschaft, sich unter gleichen Voraussetzungen gleich zu entwickeln [5]. Für den praktischen Einsatz von GAs muß aber sichergestellt sein, daß das System zuverlässig in Richtung des globalen Optimums seiner Fitnessfunktion tendiert. Konträr dazu steht die Forderung nach einer möglichst schnellen Berechnung. Der Anspruch, einen möglichst guten Kompromiß zwischen beiden Forderungen zu finden, stand daher bei der Entwicklung des GA an erster Stelle.

Tests anhand der in Kapitel 5 beschriebenen Datensätze werden im folgenden zur Evaluierung des GA herangezogen. Dabei wird eine Reihe von Parametern systematisch geändert und ihr Einfluß auf die Konvergenz der Populationen im Kontext der Faktor-Selektion untersucht.

7.1 Parameter und Stopfunktion des Genetischen Algorithmus

Die im Zusammenhang mit GAs wichtige Auslegung der Startparameter wie Generations- und Populationsgröße sowie der Auswahlwahrscheinlichkeit der Operatoren wurden anhand simulierter Datensätze (s. 5.1) und an Daten wäßriger Systeme (s. 5.2.3) getestet. Im Vordergrund der Untersuchungen stand die Konvergenz der Population und die Robustheit des GA. Es sollte ermittelt werden, inwieweit sich

die Änderung der einzelnen Parameter auf das Konvergenzverhalten und die Zuverlässigkeit der Berechnung auswirkt.

Einen weiteren wichtigen Punkt stellt die Forderung nach kurzen Rechenzeiten dar. Sie wurde durch die Implementierung eines Abbruchkriteriums für die Optimierungen, das die weiteren Berechnungen nach Überschreiten eines bestimmten Grenzwerts unterbindet, realisiert und führt zu einer deutlich verbesserten Performance des Algorithmus.

7.1.1 Konvergenz Genetischer Algorithmen

Zunächst wurde anhand der simulierten Datensätze untersucht, welchen Einfluß die Populationsgröße und die Anzahl der Generationen auf die Konvergenz des Genetischen Algorithmus hat.

Die Populationsgröße bestimmt die Zahl von Individuen in einer Generation des GA und begrenzt dadurch den Gen-Pool, aus dem sich potentielle Lösungen des Optimierungsproblems entwickeln. Mit zunehmender Populationsgröße erhöht sich der Rechenaufwand. Die Zahl der zu berechnenden Generationen bestimmt zum einen direkt die Rechenzeit, zum anderen ist die Änderung der Auswahlwahrscheinlichkeiten der einzelnen Operatoren an die Zahl der Generationen gebunden (s. 4.3.1). Die Abbildung 7.1 zeigt die Ergebnisse am Beispiel eines der simulierten Datensätze (Sim-A, s. 5.1) als Mittelwerte aus jeweils 10 Berechnungen. Die Anzahl der Generationen wurde schrittweise geändert (10, 30, 50, 70, 100, 150 und 200 Generationen). Die unterschiedlichen Populationen beinhalteten 50, 100, 150, 200 und 300 Individuen.

In der Abbildung ist die Abhängigkeit des *Last Improved Generation*-Wertes (LIG) von der Populationsgröße und der Anzahl der Generationen dargestellt. Der LIG-Wert gibt für wiederholte Berechnungen an, in welcher Generation im Mittel das beste Chromosom (Bitstring) zum ersten Mal in einer Population auftritt. Ist der Wert sehr klein, so läßt sich daraus schließen, daß eine hohe Konvergenz der Population schnell zur Bildung einer guten Lösung geführt hat. Ein hoher LIG-Wert ist dagegen häufig ein Indiz dafür, daß der GA lange in einem lokalen Optimum hängen geblieben ist und erst sehr spät — durch hohe Mutations- und Invader-Raten — eine bessere Lösung finden konnte.

Aus Abbildung 7.1 geht hervor, daß der LIG-Wert mit zunehmender Populationsgröße absinkt. Dies erklärt sich dadurch, daß die Wahrscheinlichkeit dafür, eine gute Lösung des Optimierungsproblems in einer Population zu finden, mit der Größe der Population zunimmt. Je mehr einzelne Individuen vorhanden sind, desto besser wird der Raum möglicher Lösungen („Lösungsraum“ — Menge aller möglichen Lösungen) während der Initialisierungsphase abgedeckt. Ein Zusammenhang zwischen LIG und der Anzahl der Generationen (und der damit verbundenen Veränderung der Aufrufwahrscheinlichkeiten einzelner Operatoren) läßt sich aber weder für diesen noch für andere simulierte Datensätze eindeutig feststellen. Hier ergibt sich daher der wesent-

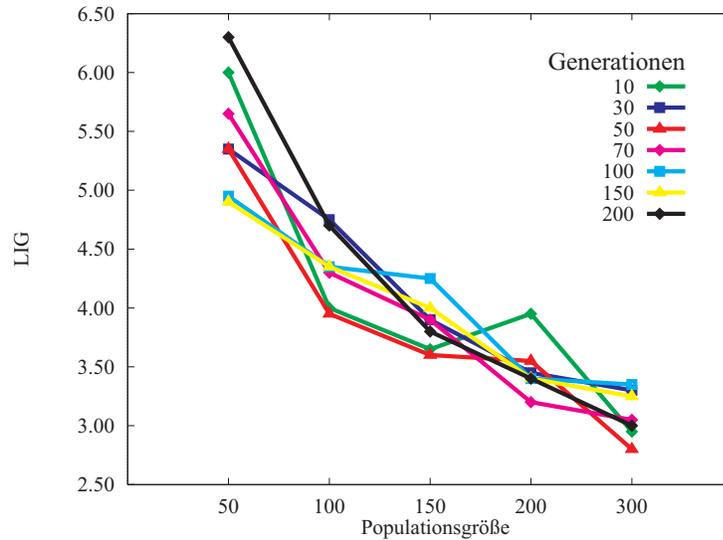


Abbildung 7.1: Änderung des LIG-Wertes in Abhängigkeit von Populationsgröße und Anzahl der Generationen am Beispiel des Sim-A-Datensatzes (Mittelwerte aus 10 Berechnungen).

liche Anteil an Verbesserungspotential, denn je kleiner die Anzahl an Generationen ist, die zur Berechnung einer Optimierung benötigt wird, desto geringer ist auch die Gesamtrechenzeit für die Lösung des Kalibrationsproblems.

Die Berechnungsergebnisse für alle simulierten Datensätze sind im Anhang in Tabelle A.10 auf Seite 166 im Detail aufgeführt. Insgesamt zeigen die Untersuchungen zu den Startparametern des GA ein in sich stimmiges Bild: Die Auslegung der Operatoren und Algorithmen bewirkt offensichtlich eine schnelle Konvergenz der Populationen. Dieser Trend läßt sich an den niedrigen mittleren LIG-Werten ablesen. Sie überschreiten in keiner der Berechnungen den Wert 8. Theoretisch könnte man also die Anzahl der zu berechnenden Generationen drastisch einschränken und so eine sehr kurze Rechenzeit erzielen. Folgende Aspekte sprechen jedoch dagegen:

1. In den Versuchen war die Anzahl der Freiheitsgrade auf 15 beschränkt. Mehr Freiheitsgrade bedeuten aber mehr Kombinationsmöglichkeiten und führen zu einer langsameren Konvergenz.
2. Die Art der gewählten Fitnessfunktion beeinflusst das Konvergenzverhalten [5] und stellt daher einen Unsicherheitsfaktor dar.
3. Sehr kleine Generationszahlen würden zu einer drastischen Veränderung der Auswahlwahrscheinlichkeiten für die Operatoren des GA führen und können sich unvorhersehbar auf die Entwicklung von Populationen auswirken.

Ein Wert von weniger als 50 Generationen hat sich bei den vorliegenden Berechnungen als kritisch erwiesen. In diesen Fällen stieg die Häufigkeit, mit der GAs nur lokale Optima erreichten, an. Die Ursache liegt in der linearen Anpassung der Wahrscheinlichkeiten, mit denen die Operatoren im GA ausgeführt werden. Wird die Anzahl maximal berechneter Generationen stark reduziert, dann sinkt die Wahrscheinlichkeit für die Ausführung des Uniform-Crossover-Operators drastisch ab (s. Tab. 4.5). Gerade dieser Operator ist aber entscheidend für die gute Konvergenz in der Anfangsphase des Genetischen Algorithmus. Letzten Endes darf auch die Tatsache, daß es sich bei Genetischen Algorithmen um komplexe Systeme handelt, die weitestgehend durch Zufallsfunktionen gesteuert werden, nicht unberücksichtigt bleiben. Bei der wiederholten Berechnung ein und desselben Datensatzes wird man daher keine absolut gleichen Lösungswege einer Optimierung finden. Diese Tatsache allein sollte schon davor abschrecken, die Rahmen-Parameter unter denen sich die Populationen entwickeln zu eng zu stecken.

Neben der Populationsgröße und der Anzahl von Generationen stellen die Auswahlwahrscheinlichkeiten der Operatoren im Genetischen Algorithmus wichtige Parameter für die Konvergenz der Populationen dar. Die Uniform-Crossover-Rate ist der zu Beginn der Optimierung dominierende Operator. Für die Beurteilung der Leistungsfähigkeit des GA ist es notwendig, abzuschätzen wie groß der Einfluß unterschiedlicher Crossover-Raten auf die Konvergenz des Systems ist. Die Abbildung 7.2 zeigt für eine Kalibrationen von NaCl und KCl in Wasser das erste Auftreten des besten Gens in einer Berechnung (LIG) in Abhängigkeit von verschiedenen Uniform-Crossover-Raten. Für die einzelnen Kalibrationen wurde der spektrale Threshold verändert. In der Kalibration werden dadurch nur Absorbitionsbereiche des Spektrums berücksichtigt, die kleiner als dieser Grenzwert sind. Ein hoher Threshold trägt zu stärkeren nichtlinearen Effekten der Kalibration bei, was entsprechend komplexere Modelle fordert.

Die Startwerte des Uniform-Crossover-Operators wurden in einem Bereich zwischen 80% und 95% variiert¹. Da die Population anfänglich sehr homogen ist, sind Werte von kleiner als 80%, die eine höhere Mutations- oder Invader-Rate mit sich bringen würden, nicht sinnvoll. Eine steigende Mutations- oder Invader-Rate ist nur sinnvoll, wenn die Population bereits konvergent ist und durch „willkürliche“ Veränderungen neues Genmaterial geschaffen werden soll. In Abbildung 7.2 sind die Mittelwerte aus jeweils zehn Optimierungen dargestellt. Die resultierenden LIG-Werte liegen hier zwischen neun und vierzehn. Auch hier zeigt sich also, daß bereits in einer sehr frühen Phase der Optimierung dominante Gene auftreten, die sich schnell durchsetzen. Die LIG-Werte sind trotz der unterschiedlichen Parameter sehr homogen verteilt. Dies zeigt, daß die Konvergenz des Algorithmus sehr stabil und von der Uniform-Crossover-Rate in den angegebenen Grenzen nahezu unabhängig ist.

¹Der Standardwert liegt bei 89%.

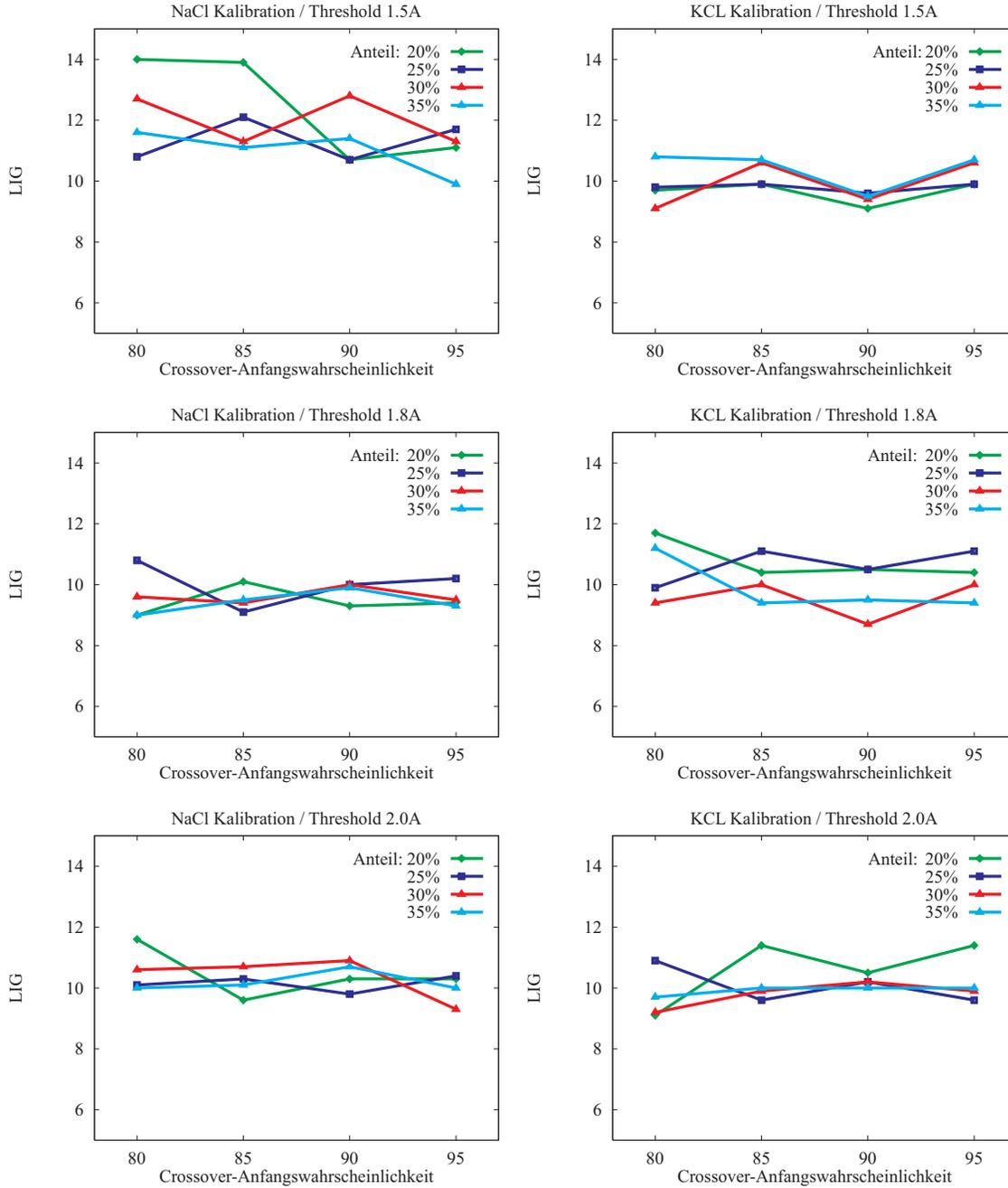


Abbildung 7.2: Abhängigkeit des mittleren LIG-Werts von der Crossover-Anfangswahrscheinlichkeit bei Berücksichtigung unterschiedlicher Anteile der besten Populationsmitglieder in der Stopfunktion. Getestet wurden NaCl- und KCL-Kalibrationen mit unterschiedlichen Thresholds für die berücksichtigten Extinktionswerte

7.1.2 Stopfunktion im Genetischen Algorithmus

Eine Möglichkeit, die Zahl der zu berechnenden Generationen trotz der zuvor beschriebenen Probleme zu reduzieren, bietet der Abbruch der Berechnung durch eine Stopfunktion. Diese muß entweder die Chromosomen einer Population (anhand ihrer Gene) oder deren Fitnesswerte evaluieren und bei Erreichen einer definierten Grenze die weitere Berechnung unterbinden. Die Stopfunktion gewährleistet, daß die Berechnung abgebrochen wird, wenn der GA mit hinreichender Wahrscheinlichkeit das globale Optimum der Fitnessfunktion erreicht hat.

Das als *Stopfunktion* bezeichnete Kriterium muß evaluieren, wie nahe eine potentielle Lösung des Optimierungsproblems (eine Faktorkombination) dem globalen Optimum kommt. Da letzteres zu Beginn der Optimierung unbekannt ist, ist dies nur näherungsweise möglich.

Der im folgenden beschriebene Lösungsansatz baut auf der Beobachtung auf, daß durch den zur Berechnung eingesetzten Algorithmus die Populationen sehr schnell konvergieren und frühzeitig das Chromosom mit der optimalen Lösung in der Population auftritt. Die Konvergenz führt dazu, daß Populationen schnell durch Chromosomen, die eine potentiell gute Lösung darstellen, dominiert werden, und der Anteil ähnlicher Lösungen bei jedem Generationswechsel stetig wächst. Es liegt daher nahe, die Varianz der Populationen in bezug auf ihre Fitness von Generationswechsel zu Generationswechsel zu vergleichen und die Berechnung abzubrechen, wenn sich keine signifikanten Änderungen mehr ergeben.

Ein Problem stellen dabei jedoch die durch die Operatoren hervorgerufenen Veränderungen in den Populationen dar. Die Operatoren sorgen in jeder Generation für neue Populationsmitglieder die fitter, aber auch weniger fit als ihre Vorgänger sein können. Die daraus resultierende unstetige Entwicklung des Fitnesswerts in der Gesamtpopulation wirkt sich störend auf die Abschätzung der Konvergenz aus. Wählt man jedoch für die Bestimmung der Varianz in der Fitness einer Generation nur einen kleinen Anteil der besten Populationsmitglieder aus, so kann man dieses Problem umgehen. Durch diese Aufteilung wird dafür gesorgt, daß die Berechnung der Varianz von Fitnesswerten zwar nicht unabhängig von den Operatoren geschieht, aber nur die generelle Tendenz der Optimierung erfaßt wird. Um sicher zu sein, daß die Berechnung nicht nach zwei aufeinanderfolgenden und zufällig sehr ähnlichen Generationen abgebrochen wird, wird dieser Vergleich über mehrere Generationen hinweg durchgeführt. Der Ablauf dieser Berechnung ist in Abbildung 7.3 dargestellt. Nach der Initialisierung des GA wird die Varianz der Fitnesswerte der besten 20 v.H. Chromosomen in der Generation ermittelt. In den folgenden Generationen wiederholt sich diese Schleife und die berechneten Werte werden mit dem Varianzwert der vorherigen Generation durch Bildung des Quotienten verglichen und in einem Register gespeichert. Ist der Mittelwert der Registereinträge über zehn Generationen hinweg kleiner als 1.00001, so kann von einer hinreichend guten Näherung des GA an das globale Optimum ausgegangen werden. Die weitere

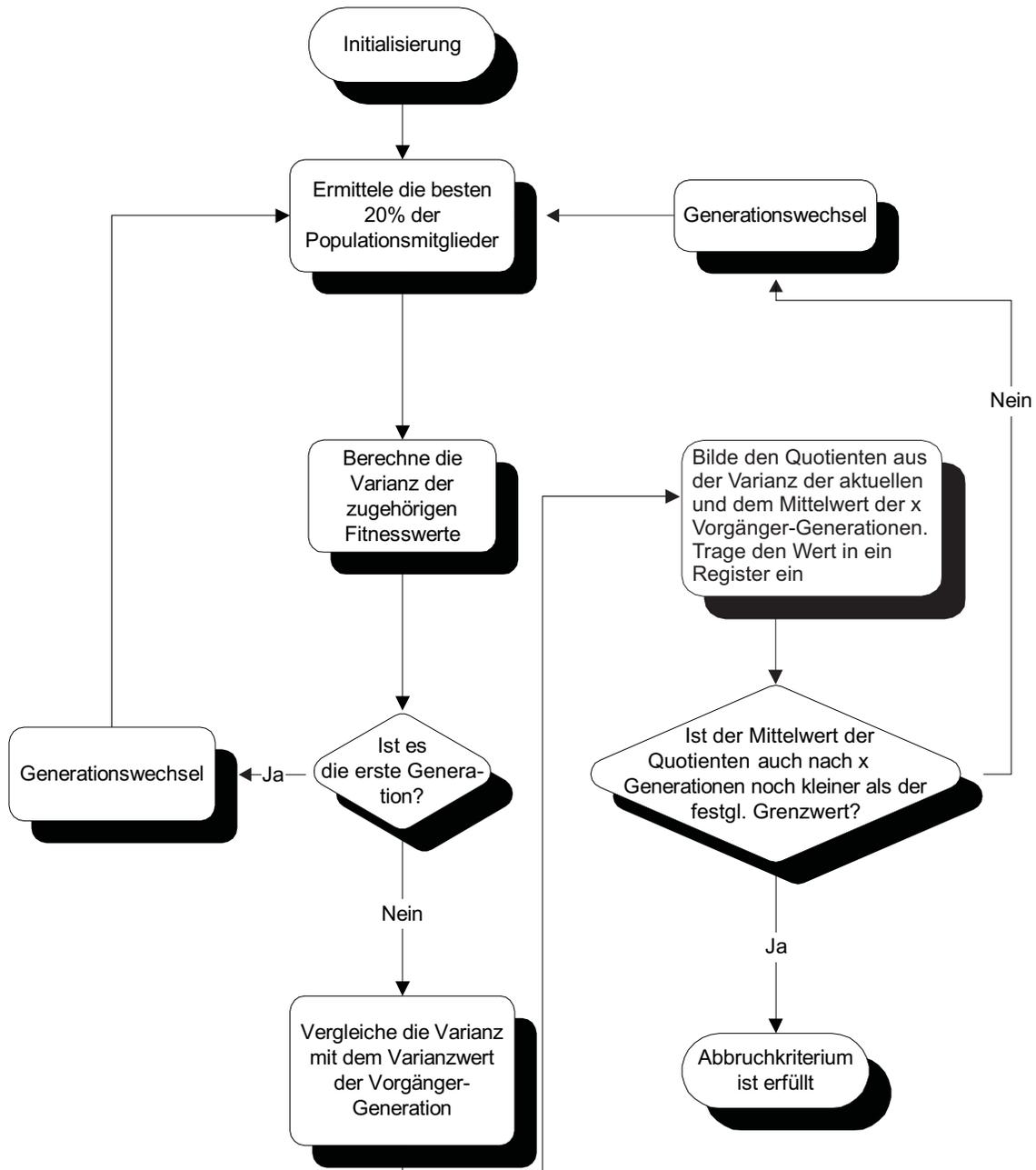


Abbildung 7.3: Aufbau der Stopfunktion im Genetischen Algorithmus.

Berechnung neuer Generationen wird abgebrochen und das beste Chromosom in der letzten Generation ermittelt und ausgegeben.

In den in Abbildung 7.2 dargestellten Versuchen wurde die Anzahl der in der Stopfunktion berücksichtigten Chromosomen in Grenzen von 20 v.H. bis 35 v.H. verändert. Es wird deutlich, daß sich daraus keine Veränderungen für die LIG-Werte ergeben.

Dagegen ändert sich jedoch die Anzahl der bis zur Auslösung der Stopfunktion berechneten Generationen (*last generation* LG). Am Beispiel der Kalibrationen von Natrium- und Kaliumchlorid-Lösungen ist dies in Abbildung 7.4 für jeweils zehn Wiederholrechnungen dargestellt. In den Berechnungen wurden maximal 300 Generationswechsel zugelassen und die Populationsgröße auf 200 Chromosomen begrenzt. Es wird deutlich, daß die Berechnungen schon nach etwa 35 bis 40 Generationswechseln durch die Stopfunktion abgebrochen werden (LG). Dadurch ergibt sich in bezug auf die Rechenzeit eine Ersparnis von fast 90%. Weiter läßt sich durch die Anzahl der in der Stopfunktion berücksichtigten Chromosomen eine weitere Reduzierung des LG-Werts erreichen. Auf Grund dieser Ergebnisse wurden daher in weiteren Berechnungen standardmäßig nur 20 v.H. Chromosomen innerhalb des Abbruchkriteriums berücksichtigt. Eine weitere Reduzierung des berücksichtigten Anteils ist nicht empfehlenswert, da sonst möglicherweise nur diejenigen guten Chromosomen einer Generation zur Berechnung herangezogen werden, die durch die Verwendung der Steady State Reproduction Technique von Generation zu Generation weitergereicht werden (s. 4.3.1).

Aus der Darstellung der Ergebnisse ist darüber hinaus auch ersichtlich, daß die Änderung der Crossover-Wahrscheinlichkeit oder des spektralen Thresholds keinen systematischen Einfluß auf den Zeitpunkt des Abbruchs der Berechnungen hat, da sowohl die LIG- als auch die LG-Werte diesbezüglich kaum eine Veränderung aufweisen. Darüber hinaus ist die Stopfunktion, wie die Ergebnisse für die unterschiedlichen Thresholds zeigen, auch weitgehend unabhängig von der spektralen Basis der untersuchten Datensätze². Sie evaluiert ausschließlich den Fortschritt der Optimierung auf Basis der Fitnessfunktion.

7.1.3 Robustheit Genetischer Algorithmen — Vergleich der GA-Ergebnisse mit dem globalen Optimum

Es gibt keine Garantie dafür, daß ein Genetischer Algorithmus in jedem Fall das globale Optimum findet. Daher soll an einigen Beispielen überprüft werden, wie nahe er diesem kommt. In Tabelle 7.1 sind beispielhaft zwei Ergebnisse dargestellt, die aus den Berechnungsreihen herausgegriffen wurden. Kalibriert wurden die bereits

²Je höher der Threshold ist, desto mehr nicht-lineare Effekte müssen in der Kalibration angepaßt werden.

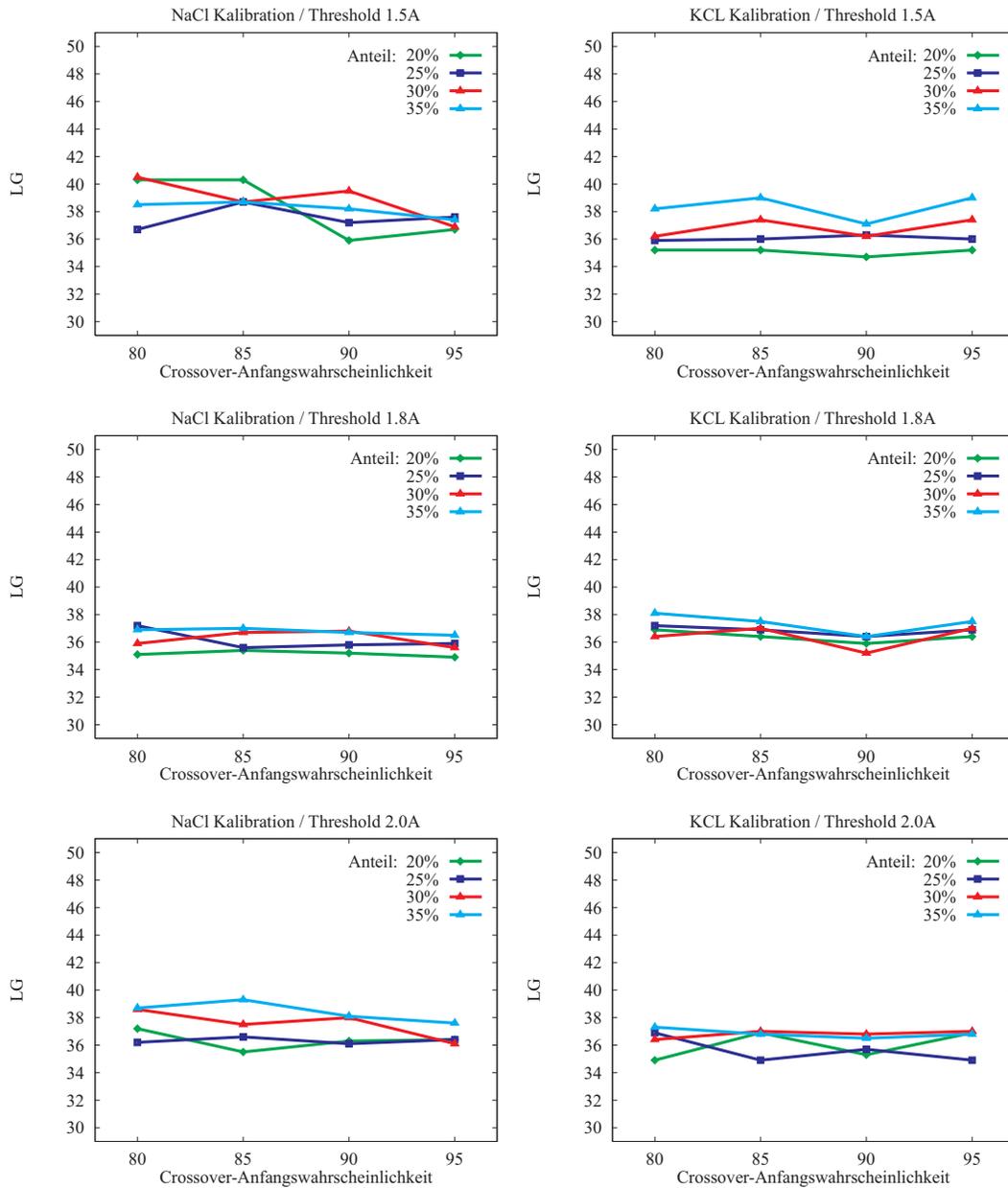


Abbildung 7.4: Abhängigkeit des mittleren LG-Werts von der Crossover-Anfangswahrscheinlichkeit bei Berücksichtigung unterschiedlicher Anteile der besten Populationsmitglieder in der Stopfunktion. Getestet wurden NaCl- und KCl-Kalibrationen mit unterschiedlichen Thresholds für die berücksichtigten Extinktionswerte

beschriebenen Natrium- und Kaliumchlorid-Lösungen mit 25 Freiheitsgraden. Die Datensätze enthalten ca. 130 Kalibrations- und ca. 60 Validationsspektren mit einer digitalen Auflösung von je 6000 Datenpunkten. Die mit NaCl und KCl-1 bezeichneten Kalibrationen wurden mit den zuvor beschriebenen Standardparametern berechnet, d.h. mit einer Populationsgröße von 200 und einer maximalen Anzahl von 300 Generationen. Die Auswahlwahrscheinlichkeit für den Uniform-Crossover-Operator betrug 90%, und 20% der Population wurden in der Stopfunktion berücksichtigt.

Tabelle 7.1: Ergebnisse des GA für zehn Wiederholrechnungen I.

| | B-XU | thr | LIG | LG | SEE | RMSEP _{V1} | SEEP | n_f |
|-------|------|-----|-------|------|----------|---------------------|----------|-------|
| NaCl | | | | | | | | |
| MIN | 90 | 1.5 | 10.70 | 35.9 | 2.24E-03 | 6.64E-03 | 4.34E-03 | 20 |
| MAX | | | 7 | 33 | 2.24E-03 | 6.64E-03 | 4.34E-03 | 20 |
| Best | | | 14 | 38 | 2.24E-03 | 6.64E-03 | 4.34E-03 | 20 |
| KCl-1 | | | | | | | | |
| MIN | 90 | 1.5 | 9.1 | 34.7 | 2.07E-03 | 2.88E-03 | 2.34E-03 | 23 |
| MAX | | | 8 | 33 | 2.07E-03 | 2.88E-03 | 2.34E-03 | 23 |
| Best | | | 11 | 36 | 2.07E-03 | 2.88E-03 | 2.34E-03 | 23 |

„thr“ steht für Threshold.

In den mit „Best“ bezeichneten Zeilen stehen die Resultate aus der Berechnung *aller* Kombinationsmöglichkeiten. Sie stellen das tatsächliche Optimum der hier verwendeten SEEP-Fitnessfunktion dar.

Sowohl bei der Kalibration „NaCl“ als auch bei der Kalibration „KCL“ findet der Genetische Algorithmus nach etwa neun bzw. elf Generationen bereits erstmals das beste Chromosom, und die Stopfunktion bricht die Berechnung nach 35 bzw. 36 Generationen ab. Jede dieser beiden Optimierungen dauert ca. 15 bis 25 Sekunden. Auf einem Dual-Pentium-Pro 200 System mit SCSI Basis und 128 MB RAM dauert die Berechnung aller 2^{25} Kombinationsmöglichkeiten dagegen etwa 24 Stunden.

Beide Kalibrationen fanden in jeder der zehn Berechnungen das globale Optimum. Nicht so jedoch die mit KCl-2 bezeichnete Kalibration (*s.* Tab. 7.2). Hier wird zwar bei der überwiegenden Zahl der Rechnungen das Globale Optimum erreicht, aber eine der zehn Wiederholrechnungen bleibt über 10 Generationen hinweg in einem lokalen Minimum stecken und ermittelt einen um den Faktor 7 schlechteren Fitnesswert. Dies kann nur aus einer ungünstigen Initialisierung der Startpopulation resultieren. Wie die Wahrscheinlichkeit für das Auftreten solcher suboptimalen Ergebnisse vermindert und zusätzlich die Konvergenz erhöht werden kann, ist Gegenstand der folgenden Abschnitte.

Tabelle 7.2: Ergebnisse des GA für zehn Wiederholrechnungen II.

| | B-XU | thr | LIG | LG | SEE | RMSEP _{v1} | SEEP | n_f |
|-------|------|-----|------|------|----------|---------------------|----------|--------|
| KCl-2 | | | | | | | | |
| MIN | 95 | 1.5 | 10.0 | 35.2 | 3.63E-03 | 4.34E-03 | 3.86E-03 | 21 3/5 |
| MAX | | | 6 | 33 | 2.07E-03 | 2.88E-03 | 2.34E-03 | 9 |
| Best | | | 11 | 38 | 1.76E-02 | 1.75E-02 | 1.76E-02 | 23 |
| | | | | | 2.07E-03 | 2.88E-03 | 2.34E-03 | 23 |

7.2 Spezielle Module des Genetischen Algorithmus

Um ungewünschte Effekte wie die zuvor beschriebenen möglichst zuverlässig zu unterbinden oder zumindest die Häufigkeit ihres Auftretens so gering wie möglich zu halten, waren eine Reihe von Weiterentwicklungen am Genetischen Algorithmus notwendig. Grundsätzlich teilen sich die daraus abgeleiteten Aufgaben in zwei Bereiche auf:

1. Methoden zu entwerfen, die schon zum Zeitpunkt der Initialisierung des Algorithmus soviel an chemometrisch relevanter Vorinformation wie möglich einbringen.
2. Bestrebungen, das Konvergenzverhalten des GA so anzupassen, daß die Wahrscheinlichkeit des Auftretens suboptimaler Lösungen minimiert wird.

Die in Punkt 1 angesprochenen Methoden werden in der Fachterminologie zu Genetischen Algorithmen als *Hybridisierung* von GAs bezeichnet. Hybridisierung dient einerseits dazu, einen Genetischen Algorithmus möglichst robust auszulegen und andererseits, für eine vorgegebene Fragestellung eine optimierte Anwendung zu liefern. Einfache GAs sind sehr robust und in der Lage, eine große Bandbreite von Optimierungsproblemen zu lösen. Aber sie berücksichtigen weder die Anforderungen der Endanwender an die Lösung einer speziellen Fragestellung noch an Rechenleistung oder das Zeitverhalten. Hybridisierung eines GA bedeutet, ihn in möglichst vielen Details genau auf das zu lösende Optimierungsproblem abzustimmen und so seine Effizienz zu erhöhen.

In diesem Kontext ist auch das Auftreten suboptimaler Lösungen des Optimierungsproblems aus Punkt 2 zu sehen. Die Steigerung von Robustheit und Konvergenz sind hier Ziel einer anderen Form der Hybridisierung. Sie soll durch die Einführung eines *Parallelen Genetischen Algorithmus* (PGA) erreicht werden. Diese Form der Erweiterung ist weniger spezifisch für ein bestimmtes Optimierungsproblem als vielmehr eine generelle Möglichkeit zur Verbesserung Genetischer Algorithmen.

Die Grundlagen zu diesen Erweiterungen und ihre Auswirkungen werden in den folgenden Abschnitten im Detail beschrieben.

7.2.1 Parallele Genetische Algorithmen

Ein wesentlicher Vorzug Genetischer Algorithmen liegt in ihrer inhärenten Parallelität, die den Charakter der simultanen natürlichen Evolution auf verschiedenen Ebenen wie Individuen, Populationen und Arten widerspiegeln kann. Die parallele Existenz unterschiedlicher Populationen und Individuen ist dabei schnell und einfach in GAs implementierbar, wobei es zwei unterschiedliche Ansätze gibt, das *migration model* und das *diffusion model* [103].

Im Falle des Migrations-Modells (s. 7.2.2) wird eine Population in sogenannte *demes*³ oder Subpopulationen aufgeteilt. Diese Subpopulationen sind unabhängig voneinander — sozusagen isoliert. Demes lassen sich am besten mit Populationen einer Art von Lebewesen auf verschiedenen Inseln vergleichen. Die meiste Zeit leben und entwickeln sich die Populationen auf den Inseln unabhängig voneinander. Doch von Zeit zu Zeit schafft es ein besonders fittes Individuum, zu einer der Nachbarinseln zu gelangen und erweitert dort den bestehenden Genpool.

In den Subpopulationen finden Selektion und Rekombination in der gewohnten Weise statt, so daß jede der Populationen zu einem eigenen lokalen Optimum konvergieren kann. Migrationen bieten die Möglichkeit zur Rekombination von Information aus einem Deme mit Information aus einem anderen Deme, so daß auch bisher unberücksichtigte Informationsanteile in die Suche nach dem globalen Optimum eingehen.

Das Migrationsmodell lebt von der Verschiedenheit der einzelnen Subpopulationen, welche durch die Migrations-Frequenz empfindlich beeinflußt wird, so daß letztere einen wichtigen Modell-Parameter darstellt. Wenn Individuen mit hoher Fitness zu häufig migriert werden, so konvergieren die einzelnen Demes gleichzeitig zum selben Optimum und verhalten sich wie *eine* große Population. Die sich daraus ergebende Gefahr ist, daß ein einzelnes Chromosom, das in einer frühen Phase der Optimierung in einer sonst homogenen Population auftritt, auch dann migriert und Dominanz erlangt, wenn es nur eine suboptimale Lösung des Problems darstellt. Findet Migration dagegen zu selten statt, geht der Nutzen des Modells vollständig verloren.

Zur umfassenden Charakterisierung von Migrationsmodellen sind noch weitere Parameter notwendig: die Zahl der Individuen, die zwischen den Demes ausgetauscht werden, die Auswahlstrategie, die zur Selektion der Individuen führt und der verwendete Ersetzungsmechanismus. Bisher gibt es zu diesem Themenbereich, abgesehen von einigen experimentellen Ergebnissen zur Migrationsrate und Anzahl der Migranten, kaum systematische Untersuchungen in der Literatur [103].

³*deme*: a group of plants or animals that are closely related and live in a single distinct locality. (biol.) Zitat aus "The Chambers Dictionary", 1993

Beim zweiten Ansatz für einen parallelen GA, dem Diffusion-Modell, findet die Parallelisierung nicht auf der Ebene von Populationen, sondern von Individuen statt. Dies erfordert die vollständige Umorganisation des Algorithmus, insbesondere im Hinblick auf die Selektionsstrategie. Gewöhnlich wird in solchen Modellen die Interaktion von Individuen auf die nächsten Nachbarn beschränkt, so daß Selektion und Rekombination „örtlich“ in dieser Nachbarschaft gebunden sind. So entsteht vorteilhafte Information (lokale Optima) an verschiedenen Stellen und wird langsam in der gesamten Population verteilt. Durch diesen lokalen Charakter ist eine Anpassung der Selektionstechnik notwendig, um eine lokale Anwendung zu ermöglichen. Darüber hinaus spielen die Größe der Nachbarschaft und die Art der Interaktion eine entscheidende Rolle, um Vielfältigkeit zu erhalten.

Auch die in Abschnitt 7.1 verwendeten nichtparallelen Algorithmen zeigten in jedem Fall eine hohe Konvergenz der Populationen. Das Problem, trotzdem in einigen Fällen nur eine suboptimale Lösung zu finden, ist in erster Linie auf drei Aspekte zurückzuführen:

1. Eine ungünstige Initialisierung der Start-Population.
2. Die hohe Dominanz einer potentiell guten Lösung, die im Verlauf der Optimierung zu einer ungewollt einseitigen Konvergenz der gesamten Population führt.
3. Die Existenz einer oder mehrerer guter Lösungen neben einem globalen Optimum mit nahezu gleichen Fitnesswerten wie dieses, aber mit sehr unterschiedlichen Chromosomen.

Der Grund dafür, daß ein Chromosom wie im 2. Aspekt aufgeführt, schnell eine Population dominiert, liegt darin, daß 'gute' Lösungen von Generation zu Generation weitergegeben (Steady State Reproduction Technique) und bevorzugt zur Rekombination herangezogen werden (Roulette Wheel Parent Selection). Ihre 'genetische' Information verbreitet sich daher sehr schnell über die gesamte Population und wird dominant. Das im 3. Aspekt angesprochene Problem führt dazu, daß der Genetische Algorithmus nicht eindeutig konvergiert, und die Stopfunktion unter Umständen nicht ausgelöst wird.

Eine Möglichkeit, diese unerwünschten Nebeneffekte ganz oder zumindest teilweise zu unterbinden, liegt in der Parallelisierung des GA⁴. In dem für die Berechnungen in dieser Arbeit eingesetzten GA wurde ein Migrations-Modell implementiert. Denn mit jedem zusätzlichen Deme im Migrations-Modell sinkt die Wahrscheinlichkeit

⁴Parallelisierung bedeutet in diesem Zusammenhang nicht, daß die einzelnen Demes auch tatsächlich parallel — auf verschiedenen Prozessoren — berechnet werden, sondern daß sie gleichzeitig nebeneinander existieren.

dafür, daß die Optimierung in einem lokalen Optimum hängenbleibt. Umgekehrt steigt jedoch mit der Populationsgröße auch die Zahl der durchzuführenden Berechnungen an, so daß im ungünstigsten Fall die Vorteile des Einsatzes Genetischer Algorithmen in bezug auf die Rechengeschwindigkeit verlorengehen können. Hier einen geeigneten Kompromiß zu finden, ist der wesentliche Punkt in den im folgenden dargestellten Untersuchungen.

7.2.2 Migrations-Modell

Die Untersuchungen zur Robustheit in Abschnitt 7.1.3 haben gezeigt, daß GAs nur in sehr seltenen Fällen⁵ dazu neigen, in lokalen Optima der Fitnessfunktion stecken-zubleiben. In einem solchen Fall würde ein Genetischer Algorithmus zu einem suboptimalen Ergebnis führen, ohne daß der Anwender sich dessen bewußt werden kann.

Läßt man die Berechnung von mehreren Genetischen Algorithmen quasi parallel laufen, so sinkt die Wahrscheinlichkeit für das ‘Hängenbleiben’ proportional mit der Zahl der eingesetzten Genetischen Algorithmen. Dies ist die Grundidee für den Einsatz des Migrationsmodells. Die einzelnen Subpopulationen bilden eine eigene „Welt“ und entwickeln sich unabhängig voneinander (vgl. Abb. 7.5).

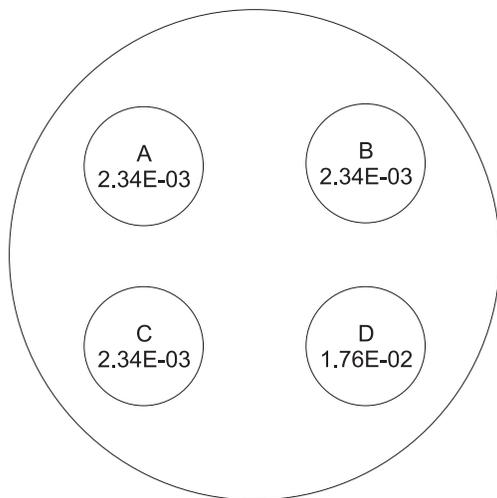


Abbildung 7.5: Einfaches Migrations-Modell am Beispiel der KCl-2 Kalibration

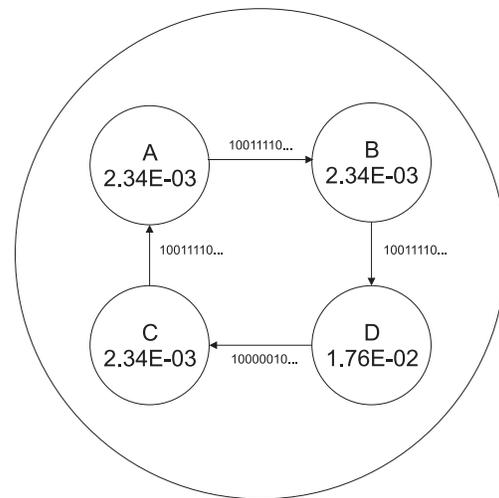


Abbildung 7.6: Erweitertes Migrations-Modell mit Genaustausch zwischen den Genetischen Algorithmen

Eine ungünstige Begleiterscheinung dieser Form des Migrations-Modells liegt darin, daß unter gleichen Vorbedingungen in jeder Generation ein kompletter Satz

⁵Insbesondere dann, wenn das zu lösende Problem einen hohen Grad an Komplexität erreicht.

von Chromosomen (Populationsmitgliedern) ausgewertet und ihre Fitness berechnet werden muß. Dadurch reduzieren sich die Vorteile der schnellen Optimierung durch GAs im Vergleich zur Berechnung aller Kombinationsmöglichkeiten. Das folgende Zahlenbeispiel verdeutlicht diesen Effekt:

Beispiel 7.1

Angenommen, ein einzelner GA braucht zur Lösung eines Problems ca. 20 Generationen, wobei in jeder Generation 200 Populationsmitglieder enthalten sind. Das bedeutet, daß ca. 4.000 mal die Auswertung der Fitnessfunktion durchgeführt werden muß. Dieser GA hat also nur dann einen Vorteil gegenüber der Berechnung aller Kombinationsmöglichkeiten, wenn die Basis des Problems größer ist als 12 ($2^{12} = 4.096$).

Verwendet man dagegen ein Migrations-Modell aus vier einzelnen, isolierten Demes bei identischen Startparametern, so müssen ca. 16.000 Berechnungen der Fitnessfunktion durchgeführt werden. Mit anderen Worten, der GA bringt erst Vorteile, wenn die Basis des Problems größer ist als 14 ($2^{14} = 16.384$). Denn es gibt keinen Grund anzunehmen, daß ein Migrations-Modell auf Basis des Modells isolierter Demes schneller konvergiert, als dies ein einzelner GA tun würde.

Wenn man jedoch, wie in Abbildung 7.6 gezeigt, einen Austausch von Informationen zwischen den einzelnen Populationen zuläßt, erlangt man einen Vorteil. Denn auf diese Weise können sich potentiell gute Lösungen im ganzen Modell verbreiten, und die Population konvergiert insgesamt schneller. Wenn es gleichzeitig möglich ist, die Anzahl der Chromosomen in den beteiligten Demes zu reduzieren, dann ist zu erwarten, daß ein Genetischer Algorithmus auf Basis des Migrations-Modells die nachteiligen Effekte in Puncto Rechenzeit aufwiegen kann.

Der Informationsaustausch kann auf verschiedenen Wegen stattfinden: Ein Weg ist die Weitergabe von zufällig ausgewählten Chromosomen (Random-Verfahren), die dann in der nächsten Subpopulation existierende Chromosomen ersetzen⁶. Dies führt dazu, daß zu jedem Zeitpunkt der Berechnung jedem GA etwa die gleiche Grundinformation zur Verfügung steht. Eine schnellere Konvergenz ist dadurch aber nicht mit Sicherheit zu erreichen.

Eine andere Strategie besteht darin, in erster Linie guten Lösungen die Migration in andere Subpopulationen vorzubehalten. Dieses Vorgehen ist mit dem Verfahren der Roulett Wheel Parent Selection bei Generationswechseln vergleichbar. Es ist zu erwarten, daß sich dadurch potentiell gute Lösungen schneller in der Gesamtpopulation verbreiten, und dies zu einer deutlich besseren Konvergenz führt.

Der bereits angesprochene problematische Teilaspekt der Dominanz eines Chromosoms tritt bei dieser Vorgehensweise jedoch wieder verstärkt in den Vordergrund. Man muß daher einen Weg finden, diesen zu vermeiden. Ein Chromosom, das nur ein

⁶Analog zur Invader-Technik oder dem Generational Replacement (s. 4.3.1).

lokales Optimum (Pseudolösung) darstellt und sich in einer sonst noch sehr homogenen Population⁷ durchsetzt, könnte sonst dazu führen, daß das ganze Migrationsmodell in Richtung eines lokalen (und nicht des globalen) Optimums konvergiert.

Die Lösung besteht darin, die Chromosomen nicht einfach auszutauschen, sondern die Interaktion mit einem Crossover-Operator zu verknüpfen. So bleibt die Information der Chromosomen nicht als Ganzes, sondern nur in Form von Teilstücken in den interagierenden Demes erhalten. Die Dominanz der einzelnen Lösungen wird erheblich geringer. In diesem Fall kann eine Pseudolösung zwar einen einzelnen Deme des Migrations-Modells dominieren, eine Ausweitung des Effekts auf die gesamte Population ist jedoch wesentlich unwahrscheinlicher.

Möglichkeiten zur Steuerung des Interaktionsmechanismus bieten sich auch durch die Verwendung von Ähnlichkeitsmaßen wie zum Beispiel der Hammingdistanz [104].

Die Berechnung der Hammingdistanz für zwei Bitstrings

$$x := (x_1, \dots, x_n), y := (y_1, \dots, y_n), \quad x_i, y_i \in \{0, 1\} \quad (7.1)$$

ist definiert als

$$d_H(x, y) := \sum_{i=1}^n |x_i - y_i|. \quad (7.2)$$

Die Hammingdistanz gibt die Anzahl der unterschiedlichen Bits in zwei Strings an und kann unabhängig vom Wert der Fitnessfunktion als Kriterium für die Ähnlichkeit von Chromosomen einer Population herangezogen werden. Auf ihrer Basis können Kriterien zur Steuerung der Interaktionen zwischen den einzelnen Genetischen Algorithmen festgelegt werden.

Im hier diskutierten Fall ist der Algorithmus so ausgelegt, daß einem Deme der Austausch von Informationen erst dann gestattet wird, wenn seine Subpopulation ein bestimmtes Konvergenzniveau erreicht hat. Dies läßt sich näherungsweise durch einen Vergleich der besten Chromosomen in den Populationen ermitteln. Ist die Population sehr homogen, dann sind die Unterschiede in den besten Chromosomen groß, da sie ein breites Spektrum des „Lösungsraums“ abdecken. Der Wert der Hammingdistanz ist entsprechend hoch. Je konvergenter eine Population wird, desto ähnlicher werden sich die besten Chromosomen, die Hammingdistanz nimmt ab. Das Potential zur Problemlösung der übertragenen Information ist zu diesem Zeitpunkt höher als beispielsweise kurz nach der Initialisierung. Die Überwachung des Migrations-Modells kann so deutlich vereinfacht werden, und die Gefahr, in einem suboptimalen Bereich der Fitnessfunktion steckenzubleiben, sinkt.

Interaktion

Sinn der Interaktion ist es, mit möglichst wenigen Berechnungen (Zeitfaktor) ein möglichst optimales Ergebnis (Güte) zu erreichen. Zu diesem Zweck wurde auf Ba-

⁷z.B. in der Frühphase der Populationsentwicklung.

sis der Hammingdistanz [104] ein Interaktionsmodul entwickelt, das die folgenden Aufgaben erfüllt:

- Anhand eines Kriteriums (*'Interaktionskriterium'*) wird festgelegt, ob ein Deme Chromosomen an andere Subpopulationen weitergeben darf oder nicht.
- Von den Subpopulationen, welche das Interaktionskriterium erfüllen, wird auf Zufallsbasis eine Liste von möglichen Interaktionspartnern erstellt und Kombinationen von Partnern werden ausgewählt.
- Ein der *Roulette Wheel Parent Selection* analoges Auswahlverfahren ermittelt Interaktions-Chromosomen.
- Ein Interaktions-Operator bestimmt die Art, in der die Informationen ausgetauscht werden.

Das Interaktionskriterium ist dann erfüllt, wenn eine bestimmte Zahl von Chromosomen in einer Generation eines Demes in einem qualitativ faßbaren Sinne konvergent ist. Dies ist der Fall, wenn die Hammingdistanz der besten fünf Chromosomen einen Grenzwert von 0.2 unterschreitet. Wie empirische Untersuchungen zeigten, führen höhere Grenzwerte nicht unbedingt zu dem gewünschten Effekt der höheren Konvergenz. Niedrigere Grenzwerte werden dagegen nur sehr selten auf Grund der im Genetischen Algorithmus eingesetzten *'steady state without duplicates'* Reproduktionstechnik erreicht. Um trotzdem für ein breites Spektrum von Anwendungen Interaktionen sicher zu gewährleisten, wurde der Grenzwert nicht statisch festgelegt, sondern analog zu den Operatoren (Crossover, Mutation, etc.) dynamisch in einem Bereich von 0.1 bis 0.6 linear an die Anzahl der berechneten Generationen angepaßt.

Die eigentliche Interaktion kann auf unterschiedliche Weise erfolgen:

1. Durch Ersatz eines Chromosoms in einer anderen Subpopulation oder
2. durch die Anwendung eines Operators (XU) auf die zur Interaktion ausgewählten Chromosomen zweier Demes.

Der eingesetzte Operator entspricht einem *Crossover-Operator*. Dies bedeutet, daß für eine Interaktion mindestens von zwei Demes der Grenzwert der Hammingdistanz unterschritten sein muß, damit ein Informationsaustausch stattfinden kann.

Die Ergebnisse in Tabelle 7.3 verdeutlichen die Effekte dieser Erweiterung auf die Robustheit und Konvergenz eines Migrations-Modells⁸. Die Berechnungsergebnisse zeigen Kalibrationen eines wäßrigen Zweikomponentensystems aus Kalium- und Aluminiumchlorid mit 52 Kalibrations- und 10 Validationsstandards, die auf Basis der SEEP-Fitnessfunktion über 300 Generationen optimiert wurden. Die Populationsgröße variiert zwischen einem Genetischen Algorithmus mit 200 Bit-Strings und

⁸Die vollständige Tabelle findet sich im Anhang A.4.

Tabelle 7.3: Übersicht über die Robustheit Genetischer Algorithmen für zehn Wiederholrechnungen.

| Property | Pop. Size | $\overline{\text{LIG}}$ | SEE | RMSEP_{V_1} | SEEP | n_f ben. | Interaktion | Wert |
|-------------------|-----------|-------------------------|----------|----------------------|----------|------------|-------------|------|
| KCL | 200 | 12.9 | 5.92E-03 | 8.67E-03 | 6.66E-03 | 18 | nein | 0.0 |
| AlCl ₃ | 200 | 84.4 | 2.21E-03 | 3.81E-03 | 2.66E-03 | 17 | | |
| KCL | 50 | 12.2 | 5.92E-03 | 8.67E-03 | 6.66E-03 | 18 | ja | 0.2 |
| AlCl ₃ | 50 | 49.6 | 2.21E-03 | 3.81E-03 | 2.66E-03 | 17 | | |

vier Demes mit jeweils 50 Bit-Strings; damit ist die Größe der Gesamtpopulation in beiden Fällen identisch.

Interessant ist hier die Entwicklung der LIG-Werte. Der mittlere Last-Improved-Generation-Wert ($\overline{\text{LIG}}$) gibt an, in welcher Generation eines Genetischen Algorithmus, das bis dahin beste Chromosom durchschnittlich⁹ zum ersten Mal aufgetreten ist. Für alle hier vorliegenden Beispiele konnte festgestellt werden, daß alle Kalkulationen zu irgendeinem Zeitpunkt das globale Optimum erreichen. Dies spiegelt sich in der Tatsache wider, daß für beide Properties in allen Berechnungen die gleichen Selektionen getroffen und daher auch dieselben Standardabweichungen (RMSEP_{V_1}) erhalten werden. Während das Migrations-Modell ohne Interaktion und großer Population verhältnismäßig häufig hohe LIG-Werte bei der Optimierung der Property Aluminiumchlorid aufzeigt, ist dies in dem Migrations-Modell mit Interaktion und deutlich reduzierter Populationsgröße nur einmal der Fall. Auch bei der Kalibration der Property Kaliumchlorid ist eine leichte Abnahme des mittleren LIG-Wertes zu verzeichnen. Der Umstand, daß die mittleren LIG-Werte kleiner werden, weist auf eine zuverlässige und schnelle Berechnung des globalen Optimums hin. Die Robustheit der GA-Optimierung ist offensichtlich größer geworden. 2Dieses Ergebnis spiegelt sich auch in den Optimierungsverläufen wider (s. Abbildung 7.7 und 7.8). Dargestellt ist die Entwicklung des Vorhersagefehlers (RMSEP_{V_1}) in Abhängigkeit von der Anzahl berechneter Generationen. Es wird deutlich, daß für Interaktionsgrenzwerte kleiner als 0.2 die Demes schlechter konvergieren und ca. 10 bis 20 zusätzliche Generationen berechnet werden müssen, bevor ein konstanter RMSEP_{V_1} -Wert erreicht wird. Umgekehrt bedeutet dies, daß durch den Einsatz des Interaktionsmoduls das globale Optimum mit weniger Berechnungen erreicht wird.

Robustheit und Konvergenz konnten, wie erwartet, gesteigert werden. Und die Berechnungen werden infolge der strukturellen Veränderungen erheblich schneller.

⁹Der Durchschnitt wird hier aus den LIG-Werten von 10 wiederholten Berechnungen gebildet.

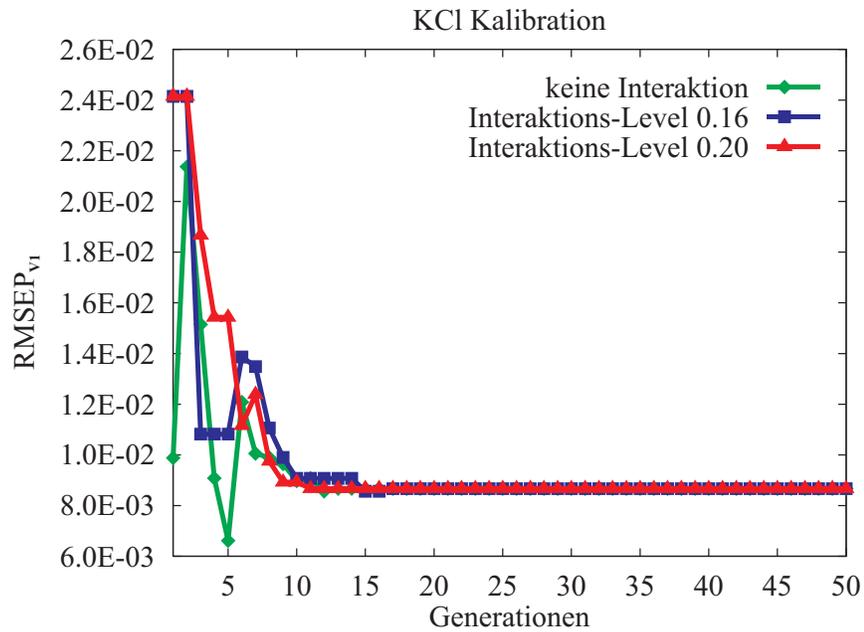
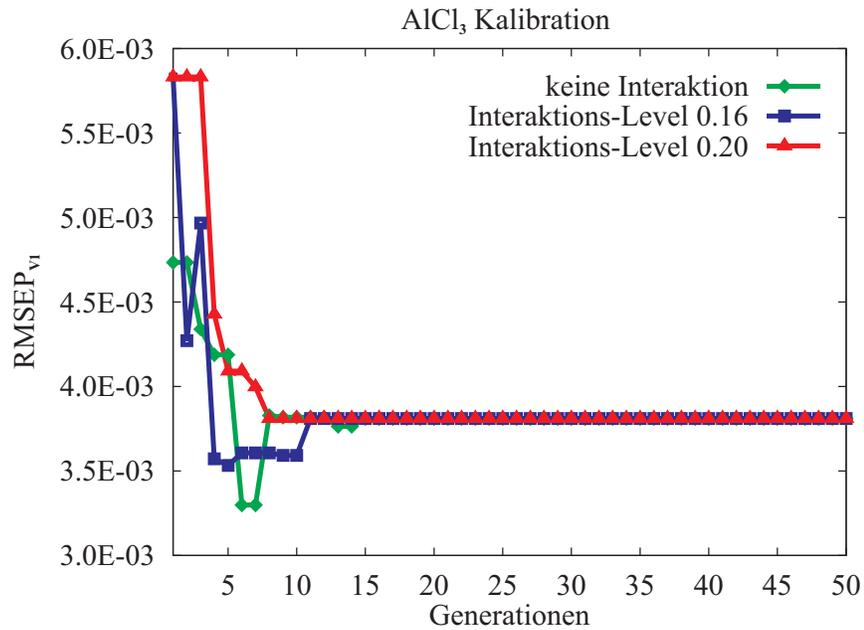


Abbildung 7.7: Konvergenz des GA (KCl Kalibration).

Abbildung 7.8: Konvergenz des GA (AlCl₃ Kalibration).