

Genetische Algorithmen und Evolutionäre Strategien

von

Adam El Sayed Auf
und
Kai Lienemann

Eine Ausarbeitung zum Seminar
„Intelligente Algorithmen“

im Wintersemester 2003/2004
an der Universität Bielefeld

Veranstalter:
Alfred Kranstedt
Nadine Leßmann
Stefan Kopp

06.04.2004

Inhaltsverzeichnis:

	Seite
1. Einleitung.....	1
2. Genetische Algorithmen	3
2.1. Erstellen der Anfangspopulation.....	4
2.2. Bewertung der Population.....	5
2.3. Selektion und Einfügen in den „mating pool“	5
2.4. Erzeugen der Nachkommen	6
2.5. Modifikationen der Teilschritte	9
2.5.1. Lösungspräsentation.....	9
2.5.2. Crossover.....	10
2.5.3. Selektion.....	12
2.6. Zusammenfassung.....	14
3. Evolutionsstrategien	14
3.1. Einführung.....	14
3.2. Codierung.....	15
3.3. Elementare Evolutionsstrategien.....	16
3.3.1. Die (1+1)-Evolutionsstrategie.....	16
3.3.2. Die ($\mu+\lambda$)-Evolutionsstrategie.....	16
3.3.3. Die (μ,λ)-Evolutionsstrategie	16
3.3.4. Selektionsdruck	18
3.4. Erweiterte Evolutionsstrategien	18
3.4.1. Die ($\mu/\rho\#\lambda$)-Evolutionsstrategie.....	18
3.4.2. Populationen.....	20
3.5. Richtwerte für die Evolutionsstrategien.....	21
3.6. Nachteile der Rechenberg-Schwefel Notation.....	22
3.7. Die Mutation	22
4. Anwendung auf das Travelling-Salesman-Problem	23
5. Vergleich zwischen ES und GA	24
6. Literaturverzeichnis	25

1. Einleitung

Als das Buch „On the Origin of Species by Means of natural Selection“ 1859 [Darwin 1859] erschien, stellte es eine Sensation dar. Nach einem Tag war die erste Auflage des Buches bereits ausverkauft. In den folgenden drei Monaten wurden drei weitere Auflagen gedruckt. Der Autor war Charles Darwin (1809-1882), dem dieses Werk Ruhm und Erfolg brachte. Kaum ein Wissenschaftler bezweifelt heutzutage noch die Richtigkeit und Korrektheit von Darwins Theorie. Sie zählt heute zu einer der wichtigsten, fundierten Theorien der modernen Biologie.

Im wesentlichen beruht Darwins Evolutionstheorie auf drei Grundannahmen und Beobachtungen. Als erste Beobachtung gilt die Tatsache, dass fast alle Lebewesen mehr Nachkommen erzeugen als überleben können. Weiterhin beobachtete Darwin, dass Lebewesen einer Art sich sehr stark ähneln können, jedoch nie vollkommen identisch sind. Es gibt also immer, wenn auch oft nur minimale, Unterschiede zwischen den Individuen einer Art. Als dritte Grundannahme sah Darwin die Tatsache, dass sich nur die erblichen Varianten in der Folgegeneration wieder finden, die sich im Kampf um Anpassung und Überleben bewährt haben. Die letzte Annahme führt langfristig, im Laufe der Generationen, zu einer Optimierung der Lebewesen einer Art.

Aus der ersten Beobachtung leitete Darwin einen „Selektionsdruck“ ab. Trotz des potentiellen Überschusses an Lebewesen, der in der Natur produziert wird, bleibt die Populationsgröße in der Regel relativ konstant. Der größte Teil der Nachkommen stirbt also, bevor er sich vermehren kann. Es entsteht schnell eine Diskrepanz zwischen Populationsgröße und lebensnotwendigen Ressourcen, wie zum Beispiel Futter. Nach Darwin werden die Lebewesen so einem „Selektionsdruck“ ausgesetzt. Dieser Selektionsdruck stabilisiert die Populationsgröße zu einer relativen Konstanz.

Darwin kombinierte seine drei Grundannahmen und leitete ab, dass bestimmte Individuen einer Art auf Grund ihrer genetischen Variation besser an ihre Umwelt angepasst sind als andere und dadurch eine größere Überlebenschance haben. Auch wenn es sich oft nur um einen geringfügigen Unterschied in ihren Anlagen handelt, halten sie dem Selektionsdruck besser stand als die weniger gut angepassten und leben lange genug um Nachkommen zu zeugen.

Es kommt also unter den Nachkommen einer Art zu einem Kampf ums Überleben. Es ist nicht unbedingt ein Kampf der Individuen gegeneinander. Der Kampf besteht sehr oft darin, mit den gegebenen Lebensbedingungen zurechtzukommen. Sei es möglichst geschickt den natürlichen Feinden zu entkommen, erfolgreich Beute zu jagen oder Futter zu finden.

Die Umweltbedingungen führen demnach zu einer natürlichen Auslese, der Selektion. Die am besten angepassten Individuen haben die größte Überlebenschance. Daher kommt auch der Ausdruck: „Survival of the fittest“. Es werden vor allem die Eigenschaften der besser angepassten Individuen an die nächste Generation weitergegeben. Auf Dauer gesehen, entwickeln sich die verschiedenen Arten auseinander und passen sich immer besser ihren Lebensräumen an.

Darwin wusste damals schon, dass eine Selektion bewusst durch Zuchtwahl erreicht werden konnte. Denn durch das gezielte Kreuzen von Individuen mit bestimmten Eigenschaften, können die Merkmale nach einigen Generationen verstärkt werden. Dieses Erkenntnis verwendet der Mensch in der Regel bei der Züchtung von Haus-, Nutztieren und Nutzpflanzen.

Aus Sicht eines Ingenieurs stellt die Evolution ein sehr interessantes und ausgefallenes Optimierungsverfahren dar. Da wir heute wissen, dass es schon vor ca. 3,5 Milliarden Jahren Leben auf der Erde gab, dauert dieser Optimierungsprozess schon eine sehr lange Zeit an und hat die erstaunlichsten Anpassungen hervorgebracht. Sogar sehr komplexe Organismen und Lebensformen können sich durch diesen Prozess in relativ kurzer Zeit auf ihre Umwelt einstellen. Die Evolution ist vergleichbar mit einem Suchprozess.

Den Suchraum, stellen dabei die genetischen Informationen bzw. der Raum der möglichen Erbanlagen dar. Das Ziel des Suchprozesses besteht in dem Finden von Erbanlagen, die dem Individuum im Kampf ums Überleben am besten dienen. Das Beeindruckende an diesem Suchprozess ist seine Effizienz.

„Bei Betrachtung der genetischen Informationen des Menschen lässt sich nach Rechenberg [Rechenberg 72] der Suchraum, den die Evolution nach optimalen Lösungen durchsuchen muss, durch einen diskreten Raum beschreiben. Die Gitterpunkte dieses Raumes stehen für alle möglichen Kombinationen von Nukleotidbasen, die im menschlichen Chromosom vorkommen können (nach Rechenberg ca. $3 \cdot 10^9$). Rechenberg schätzt demnach die Anzahl der Gitterpunkte dieses Raumes, also die möglichen Alternativen, die die Evolution potentiell durchsuchen muss, auf mindestens $4^{3.000.000.000}$. Selbst wenn diese Schätzung absolut um einen Faktor von mehreren Milliarden falsch liegt, ist das Ergebnis noch immer beeindruckend, denn ein Optimierungsverfahren, das sich wie die Evolution effizient in einem solch gigantischen Suchraum „zurechtfindet“ verdient in der Tat größte Aufmerksamkeit und Beachtung.“([Schöneburg 1994] S. 96, 97).

Die Evolution beruht im wesentlichen auf drei Prinzipien: der Mutation, der Rekombination und der Selektion. Deren Kombination ist eine geschickte Verbindung aus gerichteten und ungerichteten Suchprozessen.

Die Mutation von Genen erzeugt lediglich kleine Veränderungen und somit Varianten im Erbgut. Optimierungstheoretisch gesehen, liegt Sinn und Zweck der Mutation in der Überwindung von lokalen Optima, denn die Mutation funktioniert durch zufällige Veränderungen im Erbmaterial. Somit wird verhindert, dass sich eine Population zu schnell auf ein lokales Optimum fixiert.

Bei der Rekombination werden, Gene oder ganze Gensequenzen der Eltern kombiniert und neu zusammengesellt um ein neues Kind zu beschreiben. An welcher Stelle eine Vertauschung stattfindet, ist rein zufällig. In der Regel werden aber nahe bei einander liegende und vor allem funktional verbundene Gene seltener getrennt, als weiter auseinander liegende Gensequenzen. Die Rekombination läuft teils zufällig, teils nach statistischen Gesetzmäßigkeiten ab. Daraus folgt, dass sie zwischen einem gerichteten und einem ungerichteten Suchprozess einzuordnen ist.

Die Selektion bestimmt welche Ausprägungen der Individuen sich in der nächsten Generation verstärken und welche nicht. Sie steuert also die Evolution und legt ihren Lauf fest. Dies kann die Selektion festlegen, solange es keine Störungen der Natur gibt, wie zum Beispiel durch Klimaveränderungen, Überflutungen, Stürme, Erdbeben etc. In der Selektion verbirgt sich demzufolge ein gerichteter Suchprozess.

Die Evolution vereint aber noch mehr als nur gerichtete und ungerichtete Suchprozesse. Sowohl die Breiten- als auch die Tiefensuche sind Prinzipien der Evolution. Es ist eine annähernd optimale Kombination aus beiden zu vermuten. Diese Kombination resultiert aus dem Faktor Zeit.

Die Evolution führt ein ständiges Rennen gegen die Zeit. Passen sich die Lebewesen und Arten nicht schnell genug an, sterben sie aus. Die Zeit ist also beschränkt und muss möglichst optimal ausgenutzt werden. Dem kann die Evolution auf zwei Arten

entgegen. Entweder wird die Generationsfolge sehr kurz gehalten oder es werden so viele Individuen wie möglich zur gleichen Zeit erzeugt. Wird die Generationsfolge kurz gehalten, können sich die Individuen den Bedingungen schnell anpassen, da es eine hohe Zahl von Variationen in kurzer Zeit gibt. Wird die Anzahl der Nachkommen so hoch wie möglich gehalten, wird die Evolutionszeit minimiert, da eine viel größere Zahl von Variationen zur gleichen Zeit existiert.

Bei fast allen Arten unterscheidet sich die Kombination aus Rekombinationszeit und Rekombinationsquote und lässt vermuten, dass die Evolution für jede Art eine möglichst optimale Kombination anstrebt. Aus Sicht der klassischen Suchstrategien beschreibt diese Kombination eine Kopplung aus Tiefen- (Reproduktionszeit) und Breitensuche (Reproduktionsquote).

Im folgenden wird gezeigt, wie sich sogar die einfachsten Nachbildungen der Evolution nutzbringend einsetzen lassen. Es soll gezeigt werden, dass durch einfache Simulation und Modellierung der Natur, ihre Vorteile herausgearbeitet und zum Vorschein kommen kann und mit Hilfe der Natur erstaunliche Ergebnisse zu erzielen sind. Die Evolution vereint gerichtete und ungerichtete Suche, sowie Breiten- und Tiefensuche und kann somit als effizientes Suchverfahren betrachtet werden.

2. Genetische Algorithmen

Der Begriff der Genetischen Algorithmen wurde durch J. Holland und D. Goldberg geprägt und weist viele Parallelen zu den von Rechenberg und Schwefel entwickelten Evolutionsstrategien auf. Da jedoch schon die geringen Unterschiede im Populationswechsel ein unterschiedliches Verhalten der Algorithmen erzeugen, ist eine differenzierte Betrachtung der Algorithmen notwendig. Im weiteren soll zunächst auf die Genetischen Algorithmen eingegangen werden.

Hollands erster Ansatz für diesen Algorithmus war zunächst sehr theoretisch und wurde durch eine praxisorientiertere Verfeinerung von Rechenberg verbessert.

Die von Holland verwendete Darstellung der Individuen einer Population erfolgte durch eine Binärcodierung, worauf die hier dargestellten genetischen Operatoren angepasst sind.

Eine erste Einteilung des Algorithmus kann in mehrere unterschiedliche Schritte erfolgen, bei der im ersten Schritt zunächst ein Individuum zur weiteren Verarbeitung aus der Population ausgewählt wird und in den sogenannten „mating pool“ kopiert wird. Die Wahrscheinlichkeit für diese Auswahl beruht auf der Fitness des einzelnen Individuum, die durch die Qualität der durch das Individuum dargestellten Lösung repräsentiert wird. Diese Bewertung muss somit bei jeder Erzeugung eines neuen Individuums neu erfolgen, um dessen Fitness im Vergleich zu den anderen Individuen einer Population zu erkennen.

Diese Schritte der Auswahl und des Kopierens werden so lange iterativ angewendet, bis die Anzahl der Kopien (Kinder) der Größe der Ausgangspopulation entspricht. Auf die nun erzeugten Kinder der Ausgangspopulation werden mehrere genetische Operatoren angewendet, wodurch sich deren Repräsentation einer möglichen Lösung für das vorliegende Problem ändern sollte. Hierdurch entsteht eine Qualitätsänderung, die durch die Bewertung aller Individuen einer neu erzeugten Population neu bestimmt werden muss, nachdem die Ausgangspopulation durch die Kindpopulation ersetzt wurde.

Diese Schritte werden so lange wiederholt, bis ein vorher festgelegtes Abbruchkriterium erfüllt wird.

Der Ansatz von Holland sah jedoch keine, wie in der Natur durch die sexuelle Fortpflanzung vorkommende, Rekombination der einzelnen Individuen vor. Dieses wurde durch eine neue Implementierung des Algorithmus durch D. Goldberg dem Ansatz hinzugefügt und die Rolle von Mutationen bei den genetischen Operatoren wurde stärker abgeschwächt, um eine zielgerichtetere Suche nach guten Lösungen zu erhalten, als das es durch zufälliges Mutieren der Individuen möglich ist.

Vor der Anwendung des Algorithmus auf ein Problem, muss jedoch zuerst geklärt werden, was wir für eine Lösung unseres Problems benötigen. Um den Algorithmus genauer zu beschreiben stellen wir uns eine simple Aufgabe, in der wir die folgende Funktion minimieren wollen:

$$F = |(a-b)| + |(b-c)| + |(c-d)| + |(d-a)|$$

Hierbei ist schnell ersichtlich, dass der Minimalwert 0 erreicht wird, wenn $a = b = c = d$ gilt. Als eine Einschränkung für unser Beispiel, dürfen die Werte für die Variablen nur ganzzahlig und in einem Intervall von 0 bis 63 liegen.

Anhand dieser Problemstellung werden die folgenden Schritte des Genetischen Algorithmus genauer erklärt.

2.1. Erstellen der Anfangspopulation

Zu Beginn des Algorithmus muss eine Menge von Lösungen erstellt werden, die eine Grundlage für das weitere Verfahren darstellt. Hierbei stellen die einzelnen Lösungen Individuen und die Lösungsmenge eine Population dar, wobei die einzelnen Lösungen als Punkte in einem Lösungsraum repräsentiert werden können.

Bei diesem Initialisierungsschritt werden die Variablen für eine mögliche Lösung mit zufälligen Zahlen belegt und dessen Gleichverteilung über die Menge der möglichen Lösungen sollte hierbei so gut wie möglich erfüllt werden. Da wir neue Individuen aus der Anfangspopulation erstellen, würde eine Häufung der Lösungen um eine bestimmte suboptimale Lösung dazu führen, dass wir uns frühzeitig auf diesen Bereich des Lösungsraum konzentrieren und eine möglicherweise schon in der Population vorhandene optimale Lösung an Bedeutung verliert.

Nach der Initialisierung müssen die Individuen noch in eine Form gebracht werden, die spätere Anwendungen von genetischen Operatoren vereinfacht.

In unserem Beispiel werden wir nun die Variablen a, b, c und d durch zufällige Zahlen belegen. Die Individuen können durch einen vierdimensionalen Vektor dargestellt werden, der die binär codierten Werte der Variablen enthält. Hier zwei beispielhafte Individuen:

Dezimaldarstellung:

$I_1 : (51, 42, 63, 9, 51)$

$I_2 : (47, 45, 58, 21, 47)$

Binärdarstellung:

$I_1 : (110011, 101010, 111111, 001001)$

$I_2 : (101111, 101101, 111010, 010101)$

Eine weitere Darstellung der Lösung ist ein hintereinander schreiben der binär codierten Werte, wodurch eine Zahlenkette erzeugt wird, die als String bezeichnet wird und das Individuum repräsentiert. Die einzelnen Nullen und Einsen werden im weiteren Verlauf als Bits bezeichnet.

Hier unsere Beispielindividuen in der Repräsentation durch einen String:

I_1 : 11001110101011111001001

I_2 : 101111101101111010010101

Die Umwandlung von der einen in die andere Lösungsrepräsentation ist relativ simpel und sie bieten je nach Art der genetischen Operatoren Vor- oder Nachteile.

2.2. Bewertung der Population

Die vorhandene Population muss nun auf die Fitness der einzelnen Individuen untersucht werden, um eine spätere Selektion zu ermöglichen. Hierbei wird eine Fitness berechnet, indem die einzelnen Lösungen in unsere Zielfunktion eingesetzt und abhängig von unserer Problemstellung die Qualitäten der Lösungen bestimmt wird.

In unserem Beispiel liegt ein Minimierungsproblem vor, somit ist eine Lösung umso besser, je niedriger der durch die Lösung dargestellte Wert ist. Hier die Bewertung unserer Beispielindividuen:

Einsetzen von I_1 in F: $|(51 - 42)| + |(42 - 63)| + |(63 - 9)| + |(9 - 51)| = 126$

Einsetzen von I_2 in F: $|(47 - 45)| + |(45 - 58)| + |(58 - 21)| + |(21 - 47)| = 78$

Da in unserem Beispiel ein Minimierungsproblem vorliegt, wäre die Fitness des zweiten Individuums höher als die des ersten und es würde bei einer späteren Selektion mit einer höheren Wahrscheinlichkeit zur Übernahme in die Nachfolgepopulation ausgewählt werden.

2.3. Selektion und Einfügen in den „mating pool“

In diesem Schritt werden Individuen der Ausgangspopulation ausgewählt und in einen sogenannten „mating pool“ kopiert, der die Grundlage für spätere Mutation und Rekombination zur Erzeugung einer Nachfolgepopulation liefert. Hierbei ist zu beachten, dass wir hier Individuen durch die Auswahl nicht aus der Population entfernen, sondern nur eine Kopie in den mating pool einfügen. Daher ist es auch möglich, dass ein Individuum mehrmals zum Kopieren ausgewählt wird und somit mehrmals im mating pool vorkommt.

Die Wahrscheinlichkeit für die Auswahl eines Individuums hängt von der zuvor errechneten Fitness ab, indem sich die Auswahlwahrscheinlichkeit jedes einzelnen Individuums aus der Division der Einzelfitness durch die Summe aller Fitnesswerte ergibt.

Diese Art der Selektion wird als „fitnessproportionale Selektion“ bezeichnet, da hierbei die Auswahlwahrscheinlichkeit sich proportional zur Fitness der Individuen verhält. Zudem ist es „nicht diskriminierend“, da Individuen trotz schlechter Auswahlwahrscheinlichkeit die Möglichkeit haben für den mating pool ausgewählt zu werden. Andere Verfahren werden wir in einem späteren Kapitel ansprechen.

Wenn wir unsere Beispielindividuen als die gesamte Population ansehen, dann ergeben sich die Auswahlwahrscheinlichkeiten wie folgt:

Auswahlwahrscheinlichkeit von I_1 : $1 - (126 / 204) = 0,38$

Auswahlwahrscheinlichkeit von I_2 : $1 - (78 / 204) = 0,62$

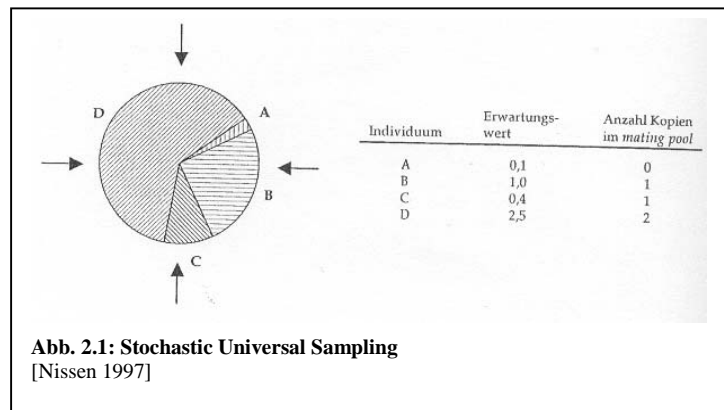
Da bei unserem Beispiel ein Minimierungsproblem vorliegt und somit ein hoher Wert gerade kein gutes Ergebnis ist und daher die Fitness bei hohen Werten schlechter wird, muss der ermittelte Wert von Eins subtrahiert werden um die korrekte Auswahlwahrscheinlichkeit zu erhalten. Somit erhalten wir das zu erwartende Ergebnis, dass das zweite Individuum eine höhere Auswahlwahrscheinlichkeit erhält und es zu

erwarten ist, dass dieses in der Nachfolgepopulation häufiger vorkommt. Hierbei sei betont, dass dies alles nur auf Wahrscheinlichkeiten beruht und es theoretisch auch möglich ist, dass Individuen mit hoher Auswahlwahrscheinlichkeit kein einziges Mal zum Kopieren ausgewählt werden, auch wenn dieses unwahrscheinlich ist. Für die spätere Betrachtung des Beispiels werden beide Individuen in den mating pool kopiert.

Um dieses besser zu verstehen, kann die Auswahl anhand eines Drehrades dargestellt werden, das so viele Einteilungen hat, wie Individuen in der Population vorhanden sind. Die verschiedenen Abschnitte repräsentieren die einzelnen Individuen und die Breite der Abschnitte entspricht ihrer Auswahlwahrscheinlichkeiten. Somit erhält ein Individuum mit einer hohen Auswahlwahrscheinlichkeit einen breiten Abschnitt und die Wahrscheinlichkeit zur Auswahl des Individuums ist somit bei einem Drehen des Rades höher, als die eines schmalen Abschnitts.

Um nun ξ Individuen auszuwählen, wird zunächst ein Punkt ausserhalb des Rades festgelegt und das Drehrad ξ -mal gedreht. Wenn das Rad nun stehen bleibt, wird das Individuum ausgewählt und kopiert, vor dessen Abschnitt der Punkt beim Stillstand des Rades liegt. Da bei diesem als „roulette wheel selection“ bezeichneten Algorithmus selbst Individuen mit sehr geringer Auswahlwahrscheinlichkeit (theoretisch) häufig zur Kopie in den mating pool ausgewählt werden können, existiert eine leichte Modifikation dieses

Verfahrens. Hierbei werden ξ Punkte gleichmäßig um das Rad herum angeordnet und das Rad nur einmal gedreht. Jedes Individuum wird nun so oft kopiert, wie Punkte vor dessen Abschnitt liegen (siehe Abb. 2.1). Dieser Auswahlalgorithmus wird als „Stochastic Universal Sampling“ (SUS) bezeichnet.



2.4. Erzeugen der Nachkommen

Der mating pool bildet nun die Grundlage für die Erzeugung einer Nachfolgepopulation, indem durch $\xi/2$ -malige Auswahl zweier Individuen und Anwendung von genetischen Operatoren (Crossover, Mutation) auf diese, veränderte Individuen erzeugt werden. Diese werden dann der Nachfolgepopulation hinzugefügt, bis alle Individuen aus dem mating pool entfernt sind.

Eine genauere Betrachtung der Einzelschritte ist notwendig, da hierbei die Variation der Genetischen Algorithmen erfolgt und somit für ein besseres Verständnis die Rolle der Einzelschritte separat betrachtet wird.

Partnerwahl

Ab diesem Schritt haben die Qualitäten der einzelnen Individuen keinen Einfluss mehr auf die Auswahlwahrscheinlichkeit, denn es werden nun aus dem mating pool zwei Individuen jeweils mit der Wahrscheinlichkeit $1/\xi$ gezogen. Somit besteht für jedes Individuum die gleiche Wahrscheinlichkeit und wenn es einmal aus dem mating pool gezogen wird, wird es nicht kopiert, sondern komplett aus diesem entfernt.

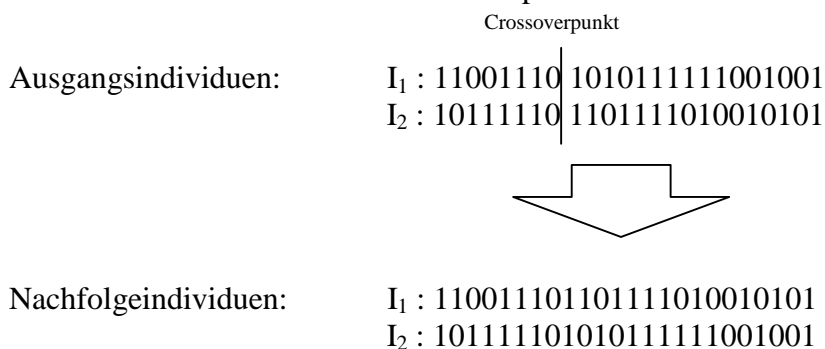
Auf den beiden ausgewählten Individuen werden nun die genetischen Operatoren Crossover und Mutation angewendet, welches wir auch mit unseren Beispielindividuen durchführen werden

Crossover

Das Crossover spielt bei den Genetischen Algorithmen die größte Rolle um neue Lösungen zu generieren. Hierbei wird bei dem Ein-Punkt-Crossover zunächst anhand der vorher festgelegten Crossoverwahrscheinlichkeit entschieden, ob überhaupt ein Crossover stattfindet oder nicht. Falls ein Crossover stattfindet wird zuvor noch stochastisch ein Crossoverpunkt bestimmt, der innerhalb des Codierungsstring der beiden Individuen liegt. Um einen Informationsaustausch zwischen den beiden Individuen durchzuführen werden nun alle Informationen (Bitbelegungen) rechts vom Crossoverpunkt ausgetauscht..

Durch dieses Verfahren werden Teile der Lösungsansätze zwischen zwei Individuen ausgetauscht und somit könnte es passieren, wenn zwei mittelmässige Lösungen Teile der Variablenbelegungen austauschen, dass eine gute und eine schlechte Lösung entsteht und somit eine Annäherung an die optimale Lösung vollzogen wird.

Dieses soll auch wieder an unseren Beispielindividuen demonstriert werden:



Hierbei werden jedoch keine neuen Belegungen der Bits erzeugt, sondern lediglich die vorhandenen zwischen den Individuen ausgetauscht. Ein Operator, der unabhängig von anderen Individuen neue Bitwerte erzeugt, wird als Mutation bezeichnet.

Mutation

Die Mutation ist bei der Suche nach optimalen Lösungen nur von nachrangiger Bedeutung, sie ermöglicht es jedoch den Lösungsraum sorgfältiger zu durchsuchen. Außerdem ermöglicht sie den Sprung aus einem suboptimalen Bereich des Lösungsraums, falls sich die Suche bei einer suboptimalen Lösung verdichtet. Da das Crossover keine Möglichkeit hat Teile einer Lösung gezielt zu verändern, sondern nur vorhandene Teile austauscht, wäre es somit unmöglich neue Lösungen in Betracht zu ziehen, falls alle Individuen die gleichen Werte besitzen.

Zunächst wird anhand einer vorher festgelegten Mutationswahrscheinlichkeit entschieden ob überhaupt eine Mutation des Individuums stattfindet oder nicht. Aufgrund der geringen Bedeutung der Mutation im Finden einer optimalen Lösung, liegt die Wahrscheinlichkeit zumeist zwischen 0,01 und 0,001.

Bei der Mutation werden einzelne Bits der Lösungen inkrementiert und somit eine Eins zu einer Null verändert und eine Null zu einer Eins, wodurch neue Werte für die Bits und somit für die Variablen entstehen.

Eine Mutation wird auch an unserem ersten Beispielindividuum durchgeführt, wobei hier die fünfte Stelle im String mutiert wird.

I_1 : 11000110101011111001001

Hierbei wird aus der Eins eine Null und die Belegung für die Variable a ergibt 49, wodurch auch eine Verbesserung des Fitnesswerts zu einem Wert von 122 entsteht. Diese geringe Änderung der Werte beruht auf der Veränderung eines für die Binärcodierung niederwertigen Bits. Wenn wir nun das erste, siebte oder dreizehnte Bit verändert hätten, hätten sich die Werte um den Wert 32 verändert, was eine gravierende Veränderung des Fitnesswerts zur Folge hätte.

Qualitätsbewertung und Zuordnung in die neue Population

In dem letzten Schritt des Algorithmus werden die Fitnesswerte der beiden durch Rekombination und Mutation veränderten Individuen neu bewertet und der Nachfolgepopulation zugeordnet.

Nach der Bewertung und dem Hinzufügen werden wieder zwei Individuen aus dem mating pool ausgewählt und die genetischen Operatoren auf sie angewandt. Dieses wird so lange fortgesetzt bis sich keine Individuen mehr im mating pool befinden und sich die Größen der Nachfolgepopulation und der Ausgangspopulation entsprechen.

Testen auf Abbruchkriterien

Die Schritte von Abschnitt 2.2.3 bis 2.2.4.4 werden so lange wiederholt bis eines der vorher festgelegten Abbruchkriterien erfüllt wird.

Ein Abbruchkriterium könnte hier eine Zeitbeschränkung für die Durchführung des Algorithmus sein, bei der ein Abbruch des Algorithmus nach einer bestimmten Zeit unabhängig von der Qualität der Lösung vollzogen wird. Diesem Kriterium ähnlich ist die Beschränkung der Generationenzahlen, so dass nach einer bestimmten Anzahl von Generationen abgebrochen wird, da davon ausgegangen werden kann, dass sich an den vorhandenen Lösungen nach einer gewissen Generationenzahl kaum noch etwas ändert. Diese Vereinheitlichung der Individuen mit steigender Generationenzahl ist auch in der Tat zu erwarten, da Individuen mit hoher Auswahlwahrscheinlichkeit öfter in den mating pool kopiert werden und sich dieser somit immer mehr vereinheitlicht. Diese Entwicklung kann auch als Abbruchkriterium verwendet werden, so dass der Algorithmus abgebrochen wird, falls es durch Rekombination kaum noch möglich ist neue Individuen zu erzeugen und die Wahrscheinlichkeit durch Mutation bessere Lösungen zu erhalten zu gering wird.

Es ist jedoch nicht immer nötig die optimale Lösung für ein Problem zu erhalten, sondern es ist eine Lösung gefordert, die eine gewisse Qualität übersteigt. In Bezug auf das Travelling Salesman Problem, das wir in einem späteren Kapitel noch genauer betrachten werden, kann dieses der Fall sein, wenn man an einer Route durch eine definierte Menge von Städten interessiert ist, die eine gegebene Länge unterschreitet. Der Algorithmus wird dann so lange durchgeführt, bis ein Individuum in der Population existiert, das eine Tour darstellt, die dieses Kriterium erfüllt und der Algorithmus kann somit in diesem Schritt abgebrochen werden und das Individuum als Lösung für die Problemstellung angesehen werden.

Die Auswahl der Abbruchkriterien hängt immer von der Art der Anwendung ab und kann in beliebigen Variationen kombiniert werden.

2.3. Modifikationen der Teilschritte

Der zuvor dargestellte Algorithmus ist eine Grundform für die Durchführung der Genetischen Algorithmen. Diese Grundform kann jedoch noch in seinen Teilschritten modifiziert werden, welches auf Grund einer speziellen Problemstellung oder aus Effizienzgründen erfolgen kann.

Eine Problemstellung, die eine Modifikation in den Teilschritten erfordert, ist das schon angesprochene Travelling Salesman Problem (TSP). In einem späteren Kapitel werden wir sehen, dass eine Darstellung durch Vektoren hierbei nicht sinnvoll ist und eine Darstellung durch Sequenzen vorzuziehen ist. Eine Sequenz ist eine Menge von Elementen und die Qualität der Sequenz hängt von der Reihenfolge und Anzahl der Elemente ab. Im Falle des TSP darf die Anzahl eines Elements in einer Sequenz nicht verändert werden, wodurch wir für Sequenzen spezielle Operatoren für Mutation und Rekombination benötigen, die sogenannten Sequenzoperatoren.

Modifikationen bieten eine grössere Auswahl an Werkzeugen, mit denen wir unsere Genetischen Algorithmen realisieren bzw. programmieren können. Ansätze für Modifikationen ergeben sich in den Teilschritten der Lösungsrepräsentation, dem Crossover und der Selektion

2.3.1. Lösungsrepräsentation

Gray Codierung

Wie bei der Besprechung der Lösungsrepräsentation durch Binärcodierung angesprochen, kann es passieren, dass sich Werte einer Lösung durch Veränderung eines einzigen Bits drastisch ändern. Dieses beruht auf der Tatsache, dass es in der Binärcodierung höherwertige Bits gibt, deren Relevanz für den codierten Wert höher ist, als die von niederwertigeren Bits.

Das beschriebene Phänomen kann durch die Gray Codierung abgeschwächt werden, indem sich Zahlen, die nah beieinander liegen, auch in ihrer Codierung geringfügig unterscheiden sollten. Die Hammingdistanz, die eine Aufsummierung aller unterschiedlichen Bitstellen von zwei zu vergleichenden Strings darstellt, sollte demnach für benachbarte Zahlen wie Neun und Zehn die Distanz Eins ergeben.

Für die Umwandlung von Binärcodierung in Gray Codierung wird folgende Regel angewandt:

$$\gamma_z = \begin{cases} a_z & \text{für } z = 1 \\ a_{z-1} \oplus a_z & \text{sonst} \end{cases}$$

Hierbei entspricht γ_z dem Bit Nummer z in dem durch Gray Codierung dargestellten String von links aus gezählt und a_z dem entsprechenden Bit in dem zu codierenden String. Der Operator \oplus entspricht hierbei der Addition modulo Zwei im Binärraum.

Um nun wieder von der Gray Codierung zur Binärcodierung zu gelangen wird folgende Formel angewandt, wobei γ_z , a_z und \oplus wie bei der obigen Formel verwendet werden:

$$a_z = \bigoplus_{k=1}^z \gamma_k$$

Die Vorteile dieser Codierung sind jedoch umstritten, da die Vorteilhaftigkeit von der Anzahl der zur Codierung verwendeten Bits abhängig ist.

Nichtbinäre Repräsentation

Wie schon zu Beginn erwähnt, gibt es neben der binären Codierung auch noch die Codierung durch Vektoren. Weitere nichtbinäre Repräsentationen stellen Codierungen durch Matrizen, Permutationen oder Baumstrukturen dar.

Die Auswahl der passenden Repräsentation hängt stark von der Art der Problemstellung ab, kann aber auch variabel während des Verlaufs des Algorithmus von einer Repräsentation in die andere überführt werden, welches jedoch in der Praxis eher selten angewandt wird.

Diploidie

Eine weitere Art der Codierung nutzt einmal mehr die Natur als Vorbild und ordnet jedem Individuum mehrere Strings zu, die für jeweils einen Lösungsansatz stehen. Diese Strings machen jedoch nur den Genotyp aus und somit nur einen Vorrat an Lösungen. Welche Lösungsteile jedoch ausgeprägt werden hängt von der Dominanz oder Rezessivität der einzelnen Bits der Strings ab.

Bei einer zeilenweisen Anordnung der Strings wird pro Spalte immer nur ein Bit für die, durch das Individuum dargestellte Lösung verwendet.

2.3.2. Crossover

Da das Crossover bei den Genetischen Algorithmen eine große Rolle spielt gibt es für diesen genetischen Operator auch Modifikationen, von denen sich die Entwickler bessere Ergebnisse und bessere Laufzeiten versprechen. Über generelle Verbesserungen kann kaum eine Aussage getroffen werden, da bei diesem Verfahren viel auf Zufall beruht und auch von der Struktur der Problemstellung abhängig ist. Einige Modifikationen sollen hier jedoch angesprochen werden, um eine Möglichkeit zu bieten, Variationen in dem Ablauf des Algorithmus für bessere Ergebnisse zu verwenden, da durch das Crossover eine globale Suche nach besseren Lösungen gewährleistet wird.

Auch bei den folgenden Varianten muss wieder zunächst anhand einer Crossoverwahrscheinlichkeit entschieden werden, ob überhaupt ein Crossover stattfindet.

N-Punkt-Crossover

Eine simple Erweiterung des zuvor vorgestellten Ein-Punkt-Crossover stellt das N-Punkt-Crossover dar, bei dem anstelle von einem Crossoverpunkt, N vorhanden sind, die auf dem String verteilt werden. Ein Informationsaustausch findet dann zwischen einem ungeraden und dem darauf folgenden geraden Crossover-Punkt auf den Elternstrings statt.

Hierbei wird gewährleistet, dass trotz Crossover weit entfernte Bits auf einem Elternstring erhalten bleiben, welches bei dem gewöhnlichen Ein-Punkt-Crossover nicht der Fall ist, da mit steigender Entfernung die Wahrscheinlichkeit für eine Anordnung der Bits auf unterschiedlichen Strings steigt. Dieses Kriterium wird oft als Bewertung von Crossover-Verfahren herangezogen und wird als „positional bias“ bezeichnet. Bei dem N-Punkt-Crossover ist der positional bias geringer als bei dem Ein-Punkt-Crossover.

Uniform Crossover

Bei diesem Verfahren wird für jedes einzelne Bit entschieden, ob ein Austausch zwischen den Elternstrings stattfindet, wobei die Wahrscheinlichkeit für solch einen Austausch in der Regel zwischen 0,5 und 0,8 liegt.

Ein positional bias ist bei Uniform Crossover nicht vorhanden, da für jedes Bit die gleiche Wahrscheinlichkeit für einen Austausch besteht.

Sequenzoperatoren

Bei dem Ansatz der Lösungsrepräsentation durch Permutationen, wobei dieses durch eine Liste realisiert wird, ist zu beachten, dass jedes Element nur in einer bestimmten Anzahl vorkommt und die Reihenfolge der Elemente für die Lösungsqualität eine Rolle spielt. Für Probleme wie das Travelling Salesman Problem (TSP) ist diese Art der Lösungsrepräsentation vorteilhaft und auch für diese Art der Darstellung sollte es möglich sein eine Art des Crossovers durchzuführen ohne ungültige Lösungen zu erzeugen. Ungültig werden die Lösungen, sobald Elemente nicht mehr in der vorherigen Anzahl vorhanden sind, wodurch sich lediglich die Reihenfolge der Elemente ändern sollte, was eine Veränderung der Lösungsqualität zur Folge hat.

Einige dieser als Sequenzoperatoren bezeichnete Verfahren werden im weiteren vorgestellt, die alle geforderten Bedingungen erfüllen.

Ein Beispiel ist das *Uniform Order-Based Crossover*, das eine Abänderung des Uniform Crossover für die Anwendung auf Sequenzen darstellt. Bei diesem Verfahren wird zunächst eine Maske erstellt, die für jedes einzelne Bit entweder eine Eins oder eine Null zuordnet. Jedes Bit mit einer Eins wird nun aus dem ersten Elternstring in den ersten Nachkommenstring an die entsprechende Stelle kopiert und alle Bits mit einer Null ebenso von dem zweiten Elternstring in den String des zweiten Nachkommen kopiert.

Alle Elemente des ersten String, die mit einer Null belegt wurden und somit nicht in den ersten Nachkommen kopiert wurden, werden in eine Liste eingefügt. Die Elemente dieser Liste werden in die Reihenfolge gebracht, wie sie in dem zweiten Elternstring vorkommen und dann nacheinander von links in die freien Stellen des ersten Nachkommenstring eingefügt. Dieses wird auch mit dem zweiten Elternstring durchgeführt, nur das hier die Elemente mit einer Eins in einer Liste gesammelt, anhand des ersten Elternstring geordnet und dann von links in den zweiten String eingefügt werden.

Somit erhalten wir zwei neue Strings, die die gleichen Elemente enthalten wie zuvor, jedoch in einer anderen Reihenfolge und somit zwei neue gültige Lösungen repräsentieren. Hierbei ist zu beachten, dass die Informationen auf die Nachkommen über die relative Reihenfolge der Elemente vererbt wird.

Ein weiterer Ansatz ist die Vererbung von Nachbarschaftsbeziehungen, auch *Edge Recombination* genannt.

Nachbarschaftsbeziehungen zwischen Elementen und ihren direkten Nachfolgern und Vorgängern lassen sich in einem Graphen durch Kanten zwischen diesen Elementen

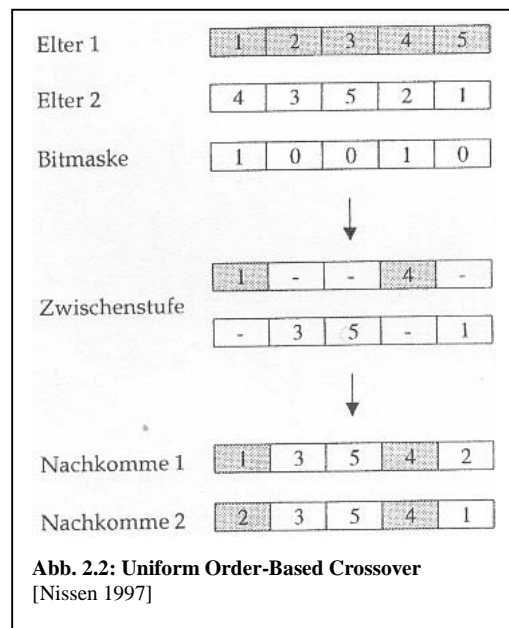


Abb. 2.2: Uniform Order-Based Crossover
[Nissen 1997]

darstellen. Hierdurch lässt sich auch der Name dieses Verfahrens erklären, welches mit Kanten-Rekombination übersetzt werden kann.

In einem Vorverarbeitungsschritt wird eine edge table erstellt, die für jeden Wert in den einzelnen Bits der Strings die direkten Nachbarn in einer Liste speichert. Hierbei werden die Werte beider Strings in dieser Liste zusammen gespeichert. Wenn nun ein Wert der Nachbar eines Wertes in beiden Strings ist und somit doppelt in der Liste vorkommen würde, wird er nur einmal als negativer Wert gespeichert.

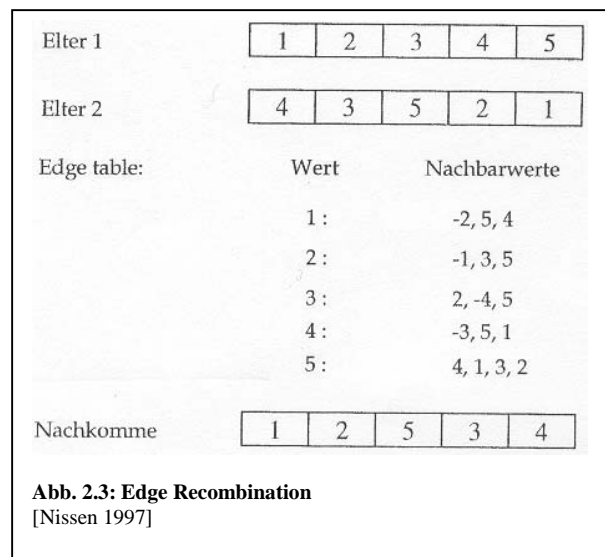
Diese edge table wird verwendet, um nun einen neuen String zu erzeugen, bei dem versucht wird, die Nachbarschaftsbeziehungen der Elternstrings beizubehalten. Zunächst wird ein Element der Strings zufällig ausgewählt und in den Nachkommen kopiert, wodurch wir nun den Startpunkt für den weiteren Algorithmus erhalten.

Nun wird für den aktuellen Wert in der edge table nachgeschaut, welche Nachbarwerte gespeichert sind. Falls ein negativer Wert existiert, wird deren Betrag gewählt und in die nächste freie Stelle in den Nachkommen kopiert. Wenn kein negativer Wert existiert, wird der Wert mit der kürzesten Nachbarschaftsliste ausgewählt und als nächster Wert in den Nachfolger kopiert. Durch die bevorzugte Auswahl von Werten, die beide Elternwerte als Nachbarn haben, wird das Erhalten von gemeinsamen Nachbarschaftsbeziehungen der beiden Elternstrings realisiert.

Nach dem Kopieren in den Nachfolger oder Auswahl als erster Wert für den Nachkommen, werden sämtliche Vorkommen des Wertes aus allen Nachbarschaftslisten gelöscht. Falls einmal mehrere Werte für eine Auswahl als zu kopierender Wert in Frage kommen, wird eine zufällige Wahl getroffen. Ein ähnliches

Verfahren wird durchgeführt, falls die Nachbarschaftsliste eines Wertes leer ist, wobei hier ein zufälliger Wert aus den anderen Nachbarschaftslisten ausgewählt wird.

Diese Schritte der Auswahl, des Kopierens und des Löschens werden so lange wiederholt bis der Nachkommenstring die Länge eines der Elternstrings hat.



2.3.3. Selektion

Für eine Besprechung des Selektionsoperators sollte dieser zunächst in zwei Teilschritte unterteilt werden und diese sollten separat betrachtet werden.

Zunächst wird durch einen Selektionsalgorithmus errechnet, wie viele Nachkommen in dem mating pool zu erwarten sind, welches durch einfache Multiplikation des Fitnesswerts eines einzelnen Individuum mit der Gesamtanzahl, der in einer Population vorhandenen Individuen, errechnet wird. Da hierbei reelle Zahlen entstehen können, jedoch nur eine ganzzahlige Menge von Individuen in den mating pool übernommen werden kann, schließt sich an diesen Schritt der Auswahlalgorithmus an, durch den die tatsächliche Anzahl von Nachkommen der einzelnen Individuen festgelegt wird.

In der Beschreibung des Grundalgorithmus wurden bereits die Selektions- und Auswahlalgorithmen *roulette wheel selection* und dessen Verbesserung das Stochastic universal Sampling besprochen. Die Verbesserung versucht die als „spread“ bezeichnete Differenz zwischen der Anzahl der zu erwartenden und tatsächlich im mating pool vorhandenen Nachkommen zu minimieren.

Als Selektionsalgorithmus wurde auch bereits die fitnessproportionale Selektion erläutert, in der die Auswahlwahrscheinlichkeit durch den Quotienten von der Fitness eines einzelnen Individuums und der Summe der Fitnesswerte aller Individuen errechnet wird. Hierbei erhalten sehr gute Lösungen eine hohe Auswahlwahrscheinlichkeit und somit ist von diesen ein grosses Vorkommen im mating pool zu erwarten. Im späteren Verlauf relativiert sich dieses jedoch durch die abnehmende Heterogenität und die Unterschiede der besten zu den schlechtesten Lösungen wird mit zunehmender Generationenzahl immer niedriger. Somit erhalten die besten Lösungen im späteren Verlauf des Algorithmus nur eine geringfügig höhere Auswahlwahrscheinlichkeit, als die der schlechtesten Lösung.

Hieraus ist zu erkennen, dass es wichtig ist, ein richtiges Verhältnis zwischen der Selektion zur Verringerung der Heterogenität und den genetischen Operatoren zu erreichen, wodurch auch in späteren Generationen ein großer Teil des Lösungsraum durchsucht wird.

Bei einer zu starken Selektion versteift sich die Suche schon sehr früh auf einen bestimmten Bereich des Lösungsraum und möglicherweise bessere Lösungen werden nicht mehr in Betracht gezogen. Bei einer zu schwachen Selektion und einer hohen Wahrscheinlichkeit für Mutation und Crossover wird es schwer zu einem endgültigen Ergebnis zu kommen und der Algorithmus verliert sein zielgerichtetes Verhalten

Rangbasierte Selektion

Bei der Rangbasierte Selektion werden die Individuen einer Population abhängig von ihren Fitnesswerten in eine Rangordnung gebracht und ihre Selektionswahrscheinlichkeit hängt ausschließlich von der Position innerhalb dieser Rangordnung ab. Somit werden relative Fitnessunterschiede zwischen den Individuen nicht mehr beachtet, sondern nur deren Stellung in der Rangordnung.

Wenn nun ein Maximierungsproblem vorliegt, hat das Individuum an erster Stelle der Rangordnung den höchsten Fitnesswert und erhält den Erwartungswert von E_{\max} Kopien und das Individuum an der letzten Stelle den Erwartungswert von E_{\min} Kopien.

Die Auswahlwahrscheinlichkeiten werden nun über folgende Formel berechnet, wobei $r(\vec{a}_i)$ den Rangplatz des Individuums \vec{a}_i darstellt:

$$p_s(\vec{a}_i) = \frac{1}{\mu} \cdot (E_{\max} - (E_{\max} - E_{\min}) \cdot \frac{r(\vec{a}_i) - 1}{\mu - 1})$$

$$\text{wobei } p_s(\vec{a}_i) \geq 0 \quad \forall i \in \{1, 2, \dots, \mu\}, \quad \sum_i p_s(\vec{a}_i) = 1$$

Da hierbei die Selektionswahrscheinlichkeit nur noch von der Rangposition der Individuen abhängt, kann der Selektionsdruck durch die Angabe von E_{\max} gesteuert werden und E_{\min} ergibt sich als $E_{\min} = 2 - E_{\max}$.

Um keine zu starke Selektion hervorzurufen wird ein Wert für E_{\max} von 1,1 empfohlen, wodurch nur schwach selektiert wird und auch zunächst schlechte Lösungen für das Kopieren in den mating pool in Frage kommen können.

Wettkampfselektion

Bei der Wettkampfselektion werden Selektionsalgorithmus und Auswahlalgorithmus in einem Verfahren vereint. Hierbei werden nun die Fitnesswerte der Individuen zunächst nicht beachtet und ξ Individuen mit der Wahrscheinlichkeit $1/\mu$ ausgewählt wobei μ die Größe der Population angibt und ξ folgende Bedingung erfüllt: $2 \leq \xi \leq \mu$

Nun wird das Individuum mit dem höchsten Fitnesswert unter den ξ Individuen in den mating pool kopiert. Somit lässt sich der Selektionsdruck bei diesem Verfahren durch

den Wert ξ steuern. Ein hoher Wert bedeutet hierbei einen hohen Selektionsdruck und ein niedriger Wert somit einen niedrigen Selektionsdruck. Falls immer nur zwei Individuen miteinander verglichen werden, spricht man von einer binären Wettkampfselektion.

Der Vorteil dieses Algorithmus liegt in der Verwendbarkeit bei negativen Fitnesswerten, da hierbei bei anderen Verfahren Probleme auftreten können, und er kann durch eine einfache Modifikation auch auf Minimierungsprobleme angewendet werden.

Als Nachteil ist jedoch die Tatsache zu nennen, dass rein theoretisch das Individuum mit der höchsten Fitness μ -mal in den mating pool kopiert werden könnte und somit der spread sehr groß werden kann, wie wir es bereits bei der Betrachtung der roulette wheel selection beobachtet haben.

Trotz dieses Nachteils wird dieses Verfahren neben der Fitnessproportionalen Selektion in vielen Anwendungen verwendet. Warum die fitnessproportionale Selektion trotz ihrer Nachteile so oft verwendet wird, kann durch eine historische Betrachtung erklärt werden. Holland und Goldberg verwendeten dieses Selektionsschema in ihren Erläuterungen zu den Genetischen Algorithmen und somit wurde es bei vielen Implementationen beibehalten.

Elite-Selektion

Das Verfahren der Elite-Selektion kann als weiterer Schritt neben anderen Selektionsalgorithmen angewendet werden. Hierbei werden die v ($1 \leq v \leq \mu$) besten Individuen einer Population auf jeden Fall in die Nachfolgepopulation übernommen. Dadurch wird versucht zu vermeiden, dass sehr gute Lösungen verloren gehen, welches durch den stochastischen Charakter der Selektion möglich wäre. Ein Nachteil hierbei liegt jedoch darin, dass bei einer Übernahme vieler Individuen durch die Elite-Selektion die Suche innerhalb des Lösungsraums auf einen suboptimalen Bereich begrenzt wird.

2.4. Zusammenfassung

Hiermit ist nun eine Grundlage geschaffen worden, mit der Problemlösungen durch Genetische Algorithmen erzeugt und Variationen in den einzelnen Teilschritten verwendet werden können. Welche Kombination von Lösungsrepräsentation, Crossover und Selektion für ein bestimmtes Problem die Beste ist, kann in den meisten Fällen nicht vorhergesagt werden und auch Parameter für die einzelnen Schritte müssen experimentell beobachtet werden. Wie in der Selektion gezeigt, haben jedoch gewisse Verfahren Nachteile, wodurch diese seltener verwendet werden sollten. Generelle Aussagen über die Effektivität der Modifikationen kann durch deren stochastischen Charakter kaum getroffen werden. So kann in einem Durchlauf ein gutes Ziel erreicht werden, in einem zweiten jedoch ein schlechteres, wodurch kaum Aussagen über die generelle Lösungsqualität der durch dieses Verfahren zurück-gelieferten Lösungen getroffen werden kann.

3. Evolutionsstrategien

3.1. Einführung

In den sechziger Jahren entwickelte Ingo Rechenberg an der Technischen Universität in Berlin ein Modell der Evolution, dass die Basis der Evolutionsstrategien bildet. Rechenberg veröffentlichte das Buch „Evolutionsstrategie“ [Rechenberg 1973].

Seit dessen Erscheinen verbesserten sowohl Rechenberg selbst, als auch andere Wissenschaftler sein Modell der Evolution. Hier sind vor allem die Arbeiten von Hans-Paul Schwefel aus den 70er und 80er Jahren von einflussreicher Bedeutung.

Die Vielzahl von Erweiterungen der Evolutionsstrategien führte zu einer interessanten Entwicklung derselben. Während die ersten Evolutionsstrategien, die im folgenden auch mit ES abgekürzt werden, noch sehr abstrakt und wenig „naturgetreu“ waren, führte die Entwicklung der ES zu immer detaillierteren Modellen. Rechenberg und Schwefel orientierten sich immer stärker an den biologischen Zusammenhängen und verliehen ihren Modellen immer mehr Naturtreue. Trotzdem vertrat Rechenberg die Meinung, dass eine komplett naturgetreue Kopie der Evolution nicht sinnvoll sei.

„Nun ist es auch falsch zu glauben, ein natürliches Vorbild müsse möglichst genau kopiert werden, um höchste Vollkommenheit in der technischen Ebene zu erreichen. [...] Das biologische Vorbild beschreibt einen Anfangspunkt, von dem aus eine Weiterentwicklung unter den speziellen technischen Bedingungen einsetzen kann. Das gilt auch für die Nachahmung der biologischen Evolution. Es muss nicht unbedingt am besten sein, jeden Evolutionsfaktor genauestens zu kopieren. Das Verstehen des jeweiligen biologischen Vorgangs kann ebenso wertvoll sein, wenn es darauf gelingt, ein idealisiertes Schema zu entwerfen, das die gleiche Wirkung hervorbringt.“ ([Rechenberg 1973], S. 15,16).

Die hauptsächlichen Unterschiede der vielen ES-Varianten liegen in den Modellierungsdetails. Rekombination, Zusammenfassung, Mutation und Selektion sind die evolutionären Operatoren. Durch ihre Vielzahl an Varianten entstehen unzählbar viele Unterschiede. Das Vorkommen so vieler verschiedener Varianten und Ausprägungen der ES ist auch der Grund für die Ausgangsfragen jedes Entwicklers von Evolutionsstrategien:

Wie detailliert sollte das Evolutionsmodell sein? Bedeutet der Verzicht auf bestimmte Details auch gleichzeitig das Einschränken der Erkenntnisfülle? Ist im Umkehrschluss auch eine „Überrepräsentation“ der Evolution durch zu viele überflüssige Details möglich?

Allgemeingültige Antworten auf diese Fragen gibt es nicht. Wie auch später noch des Öfteren erwähnt, sind die meisten Variablen hier stark problemabhängig.

Die ES verwenden eine etwas andere Codierung von Individuen als die Genetischen Algorithmen. Nach Erläuterung der Codierung, werden die wichtigsten elementaren ES vorgestellt. Im Anschluss sollen mit Beispielen aus der Natur die Grundideen der ES eingeführt und verständlich gemacht werden. Nach der Vorstellung von Richtwerten und Kritik an der hier vorgestellten Notation, wird abschließend auf die Mutation in den ES eingegangen. Der folgende Text orientiert sich sehr stark an [Schöneburg 1994]

3.2. Codierung

Aus Sicht der Evolutionsstrategen, ist die relevante Erbinformation eines Individuums in reellen Zahlen codierbar. So wird ein Individuum durch einen Vektor reeller Zahlen dargestellt und eine Population durch eine Menge solcher Vektoren.

Dieser an Realzahlen orientierter Ansatz hat historische Gründe. Denn Ingo Rechenberg beschäftigte sich als Ingenieur besonders mit der reellen Parameter-Optimierung. Das heißt, dass auch die Lösungen die er suchte in der Regel aus reellen Zahlen bestanden. Auch wenn für Rechenberg und seine Anhänger Vektoren reeller Parameter die kompakteste Codierung darstellen, ist diese Form nicht immer die zweckmäßigste. So ist zum Beispiel bei diskreten Optimierungsverfahren oder bei der Suche nach optimalen Strukturen die binäre Codierung meist vorteilhafter als die reelle Codierung. Es gibt keine problem-invariante optimale Codierung [Schöneburg 1994].

3.3. Elementare Evolutionsstrategien

3.3.1. Die (1+1)-Evolutionsstrategie

Rechenberg setzte 1994 die so genannte „zweigliedrige“ (1+1)-ES ein. Sie gilt als die einfachste Form der ES. Rechenberg versuchte mit ihr die Einstellung von Gelenkwinkeln einer Gelenkplatte zu optimieren.

Die (1+1)-ES ist so zu verstehen, dass von einem Ur-Individuum ausgegangen wird, welches genau ein Kind zeugt. Hierzu wird von dem Ur-Individuum bzw. dem Elter eine Kopie gemacht und diese im Anschluss zufällig verändert. Dabei entspricht die Duplikation des Elters dem biologischen Prozess der DNS-Selbstverdopplung und die zufällige Modifikation der Kopie entspricht dem biologischen Prozess der Mutation.

Mit Hilfe einer Qualitätsfunktion wird nun jedem Individuum ein Wert zugewiesen, der das Individuum (bzw. den Vektor) bewertet und ihm eine Qualität gibt. Hier tritt die Selektion, im Allgemeinen auch als „survival of the fittest“ bekannt, in Kraft und das bessere der beiden Individuen „überlebt“. Haben beide Individuen den gleichen Qualitätswert, so wird zufällig eines der beiden ausgewählt.

Kurz gesagt gibt es drei große Schritte: Duplikation (auch Selbstreproduktion), Mutation und Selektion. Diese drei Schritte werden so oft wiederholt, bis eine zufriedenstellende Lösungsqualität erreicht ist. Wann eine Lösung zufriedenstellend ist, hängt von der Art des Problems ab.

3.3.2. Die $(\mu+\lambda)$ -Evolutionsstrategie

Die $(\mu+\lambda)$ -ES ist eine Verallgemeinerung der (1+1)-ES. Bei dieser Strategie werden aus μ Eltern genau λ Kinder erzeugt. Beispielsweise stellt $\mu = 6$ und $\lambda = 12$ eine (6+12)-ES dar.

Bei dieser Evolutionsstrategie ist allerdings darauf zu achten, dass $\lambda \geq \mu \geq 1$ gilt. Hier werden zur Erzeugung von λ Nachkommen λ Eltern aus den μ Eltern zufällig ausgewählt. Da die Auswahl mit gleicher Wahrscheinlichkeit jedes Individuum aus der Elternpopulation trifft, ist eine Mehrfachauswahl eines Elter nicht nur möglich, sondern bei einer $\lambda > \mu$ Situation sogar notwendig.

Nach Duplikation der λ Eltern, werden die Kopien mutiert. Im Anschluss an die Bewertung durch eine Qualitätsfunktion, wie bei der (1+1)-ES, werden die λ Duplikate zusammen mit den μ Eltern in einen Pool gegeben. Es werden die μ besten Individuen aus diesem Pool ausgewählt, um die nächste Elternpopulation zu bilden. Das bedeutet, dass die Anzahl der Eltern in jeder Generation gleich und somit konstant ist.

Bei der $(\mu+\lambda)$ -ES gibt es einen ganz besonders bedeutenden Effekt, der durch die Auswahl jeder neuen Generation aus den μ Eltern und den λ Nachkommen resultiert. Da jedes Mal die besten Individuen aus sowohl den Eltern als auch den Nachkommen ausgewählt werden, kann die Qualität der besten Individuen von Generation zu Generation sich nicht verschlechtern.

Eine einprägsame und bündige Kurzformel für die $(\mu+\lambda)$ -ES ist:

μ Eltern erzeugen λ mutierte Nachkommen; die μ besten überleben

3.3.3. Die (μ,λ) -Evolutionsstrategie

Die Regelung bei der (μ,λ) -ES scheint sehr sinnvoll, da somit die besten gefundenen Individuen in jedem Fall weiterleben. Eine durch ein sehr lange lebendes Individuum repräsentierte Lösung, lässt die Evolutionäre Strategie allerdings frühzeitig zu einer Konvergenz tendieren. Stellt das langlebige Individuum jedoch lediglich ein lokales

Optimum dar, so Konvergiert die Strategie auch frühzeitig gegen genau dieses lokale Optimum.

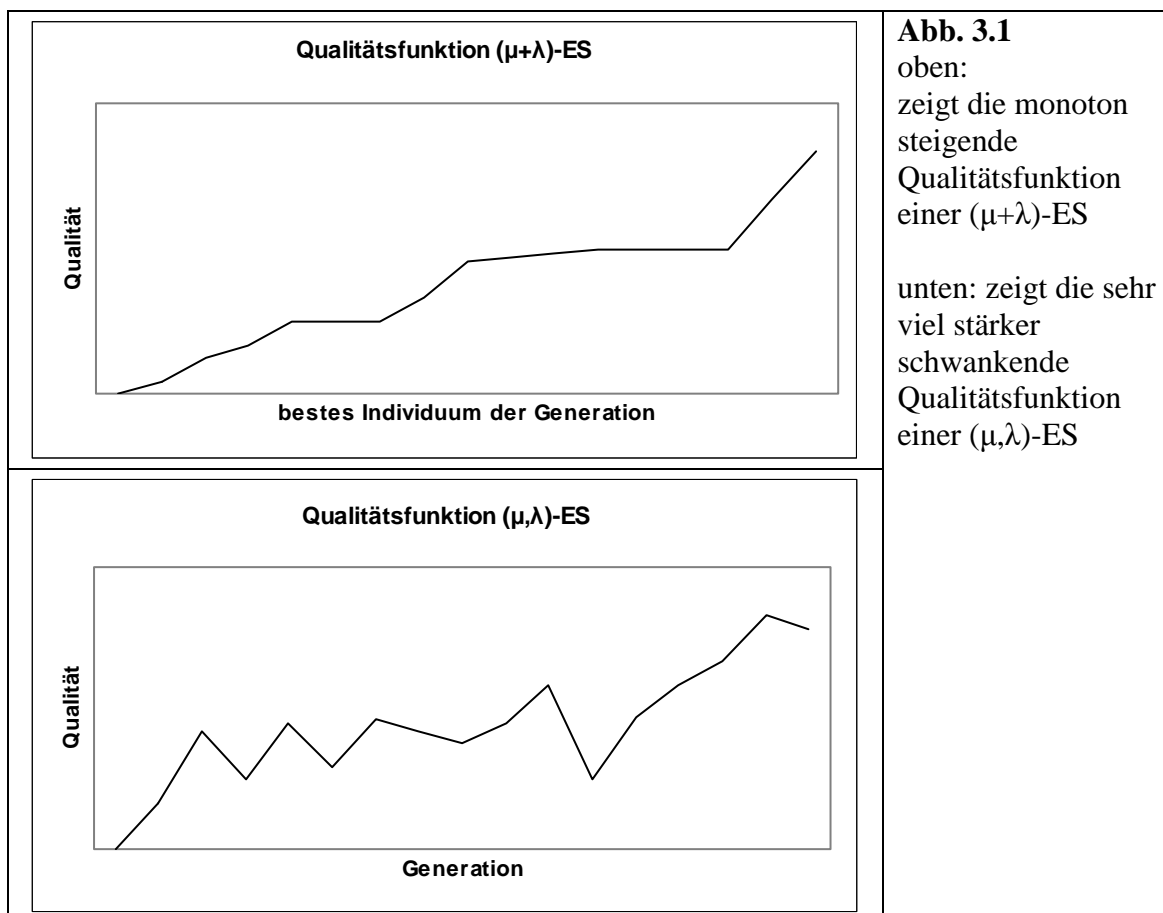
In seiner Dissertation führte H. P. Schwefel 1975 die so genannte Komma-Notation ein. Im Unterschied zur $(\mu+\lambda)$ -ES wird bei der Komma-Notation die nächste Generation ausschließlich aus den λ Nachkommen gebildet.

Mit dem Ersetzen des „+“ Symbols, durch das „,“ Symbol veränderte Schwefel das Selektionsverfahren. Da bei der Komma-Notation die Nachfolgeneration aus den λ Kindern ausgewählt wird, kommt keines der noch so guten Eltern in der nächsten Generation vor. Die (μ,λ) -ES ist somit ein wenig näher an der Natur modelliert, als die $(\mu+\lambda)$ -ES. Es gibt keine Möglichkeit mehr für ein Individuum unsterblich zu werden. Da in der Realität Individuen mehrere Generationen überleben können, unterscheidet sich die (μ,λ) -ES in diesem Punkt zwar noch immer von der Natur, ist aber intuitiver als eine Welt mit unsterblichen Individuen.

Bei Vergleich der Qualitätsfunktionen von der $(\mu+\lambda)$ -ES und der (μ,λ) -ES fällt jedoch ein potentieller Nachteil der (μ,λ) -ES auf.

Die Qualitätsfunktion einer ES zeigt die Lösungsqualität in Abhängigkeit von den besten Individuen. Hierzu ist auf der Ordinate die Lösungsqualität und auf der Abszisse das jeweils beste Individuum einer Generation aufgetragen.

Die Qualitätsfunktionen der beiden beschriebenen ES-Varianten zeigen einen sehr unterschiedlichen Verlauf. Während die Qualitätsfunktion der $(\mu+\lambda)$ -ES monoton steigt, treten bei der Qualitätsabhängigkeit der (μ,λ) -ES durchaus Schwankungen auf (Abb. 3.1).



Die Ursache für die unterschiedlichen Verläufe liegt auf der Hand. Da bei der $(\mu+\lambda)$ -ES das beste Individuum durchaus auch aus der Elterngeneration stammen kann, ist eine Verschlechterung des Besten in der nächsten Generation nicht möglich. Das beste

Individuum bleibt im schlimmsten Fall also von Generation zu Generation gleich. Eine Verbesserung ist jedoch jeder Zeit möglich. Im Gegensatz dazu ist bei der (μ, λ) -ES auch eine Verschlechterung des Besten möglich. Die Möglichkeit eines evolutionären „Rückschrittes“ ist somit nicht ausgeschlossen.

Durch das Unterdrücken groß ausfallender Mutationen können starke Schwankungen der Qualitätsfunktion aber gering gehalten werden. Eine gering ausfallende Mutation bedeutet, dass sich die Mutanten von den Originalen nur leicht unterscheiden. Somit wird eine sprunghafte Veränderung zwar nicht ausgeschlossen, aber sehr unwahrscheinlich.

Die Mutationsmechanismen der ES sind auch aus diesem Grund in der Regel klein ausfallende Mutationen.

Da sich viele Erweiterungen der ES sowohl auf die $(\mu + \lambda)$ -ES als auch auf die (μ, λ) -ES beziehen, können diese beiden Strategien auch zusammengefasst werden. Ist es nicht von Bedeutung um welche der beiden Es-Varianten es sich handelt, können sie unter $(\mu \# \lambda)$ -ES zusammengefasst werden. Das „+“ und das „ , “ verschmelzen somit zu dem „#“ Symbol.

3.3.4. Selektionsdruck

Der Selektionsdruck wird durch den Quotienten $s = (\mu / \lambda)$ errechnet. Der Selektionsdruck innerhalb einer Population ist umso stärker, je kleiner der Quotient ist. Das bedeutet, wenn λ viel größer als μ ist, findet eine starke Selektion statt. Denn die folgende Generation wird bei der $(\mu \# \lambda)$ -ES aus den nur viel wenigeren μ besten Individuen bestehen. Es gibt also viel mehr Nachkommen als in die nächste Generation übernommen werden können. Wird μ ungefähr genau so groß gewählt wie λ , gibt es nahezu keinen Selektionsdruck. In einem solchen Fall sortiert die Qualitätsfunktion die Anzahl der kleinen Differenz von λ und μ aus. Der Selektionsdruck schwankt also zwischen den beiden Extremwerten Null und Eins. Null bedeutet einen extrem hohen Selektionsdruck und Eins einen schwachen bzw. gar keinen Selektionsdruck.

Durch passende Wahl des μ - und λ -Wertes ist nun eine beliebig genaue Einstellung des Selektionsdrucks zwischen den beiden Extremwerten zu erreichen. Auch eine systematische Selektionsdruckvariation und deren Auswirkungen auf des Ergebnis ist so zu untersuchen.

Noch weitergehend können Populationswellen mit gleich bleibendem oder sich verändernden Selektionsdruck simuliert werden. So simuliert zum Beispiel eine beliebige Veränderung von μ bei gleich bleibenden λ -Wert, Populationswellen mit schwankendem Selektionsdruck. Wird μ vergrößert, wächst die Population bei sinkendem Selektionsdruck. Genau diese Situation ist in der Natur bei so genannten Nischenbildungen zu finden. Durch den Einsatz von mathematischer Funktionen sind auch zyklische oder periodische Veränderungen des Quotienten s erreichbar.

Liegt das Ziel hingegen in der Simulation von Populationswellen mit gleich bleibendem Selektionsdruck, müssen μ und λ bei jeder Variation im gleichen Verhältnis verändert werden.

3.4. Erweiterte Evolutionsstrategien

3.4.1. Die $(\mu / \rho \# \lambda)$ -Evolutionsstrategie

Wird das bisherige Rekombinationsverfahren mit einem Blick in die Natur verglichen, ist ein Verbesserungsvorschlag offensichtlich. Die angewendete Rekombinations-Methode der $(\mu \# \lambda)$ -ES, bei der aus einem Elter ein Kind gezeugt wird, ist nicht

vergleichbar mit der sexuellen Rekombination. Aus diesem Grund wird hier eine weitere Variante eingeführt.

Die $(\mu/\rho/\lambda)$ -ES basiert auf den (μ/λ) -ES und ist um das Element ρ erweitert. Das Element ρ bezeichnet in diesem Fall eine Gruppengröße. Die Größe beschreibt eine Gruppe aus der ein Kind gezeugt wird. Im Standardfall gilt $\rho=2$. Das bedeutet, dass jeweils zwei Elternteile ein neues Kind zeugen. Ähnlich wie bei der sexuellen Rekombination entsteht bei dieser Variante ein Kind aus zwei Eltern solange $\rho=2$ gilt. Wird ρ ein höherer Wert zugeteilt, entspricht es nicht mehr ganz der sexuellen Rekombination, basiert aber auf dem gleichen Grundgedanken. Die Grundidee, die dahinter steht, ist die Mischung der Erbinformationen mehrerer Individuen.

Da es sich bei den Individuen um reellwertige Vektoren handelt, stellt sich die Frage, wie aus im allgemeinen Fall n Vektoren ein neuer Vektor erzeugt werden kann.

Zur Lösung dieses Problems werden vor allem zwei Verfahren angewendet:

Erstes Verfahren:

Jeder Parameter des neuen Vektors entspricht dem Mittelwert aller Elternvektorparameter an gleicher Position. Wird von $\rho=2$ ausgegangen, so wird aus zwei Elternvektoren ein neuer Kindvektor erzeugt. Der aus den ersten Vektorparametern beider Eltern gebildete Mittelwert stellt den ersten Parameter des Kindvektors dar. Bei dem zweiten Parameter des Kindes wird genauso vorgegangen und so weiter.

Beispiel: hier gilt $\rho=2$

Elter 1: $\langle 3, 4, 6, 2, 5, 9, 2 \rangle$

Elter 2: $\langle 6, 4, 2, 4, 5, 7, 5 \rangle$

Mittelwertberechnung: $\frac{3+6}{2}, \frac{4+4}{2}, \frac{6+2}{2}, \frac{2+4}{2}, \frac{5+5}{2}, \frac{9+7}{2}, \frac{2+5}{2}$

Kind: $\langle 4.5, 4, 4, 3, 5, 8, 3.5 \rangle$

Der so gebildete Vektor wird im folgenden Schritt mutiert und das Kind ist erzeugt. Wird $\rho>2$ gewählt, ist der Mittelwert aller ρ Parameter zu bilden. Für den Fall einer höheren Wahl von ρ sprechen die Evolutionsstrategen auch von einer Multirekombination.

Zweites Verfahren:

Hier findet eine vollständige diskrete Vertauschung der Vektorparameter statt. Die Parameterwerte werden also unter den Vektoren per Zufall auf den entsprechenden Positionen vertauscht und bilden somit einen neuen Vektor. Für den Fall $\rho=2$ wird per Zufall entschieden, welcher der beiden Parameter an erster Stelle der Elternvektoren den ersten Parameter des Kindvektors entsprechen soll. Dies kann zum Beispiel mit Hilfe einer zufällig erstellten Maske umgesetzt werden. Besteht die Maske aus Nullen und Einsen, bekommt das Kind an jeder Position, die in der Maske mit Eins belegt ist, den Parameter von Elter 1 und an jeder mit Null belegter Stelle den Parameter von Elter 2.

Beispiel: es gilt $p=2$

Elter 1: $\langle 3, 4, 6, 2, 5, 9, 2 \rangle$

Elter 2: $\langle 6, 4, 2, 4, 5, 7, 5 \rangle$

Maske: $\langle 1, 1, 0, 1, 0, 0, 1 \rangle$

Kind: $\langle 3, 4, 2, 2, 5, 7, 2 \rangle$

3.4.2. Populationen

Da die Evolution sich nicht nur auf einzelne Individuen beschränkt, sondern in der Regel auf ganze Populationen, ist die Erweiterung der ES für ganze Populationen leicht nachvollziehbar. Es soll die Entwicklung unterschiedlicher und eventuell konkurrierender Populationen simuliert werden. Eine derartige Simulation bietet sich vor allem auf parallelen Rechnerstrukturen an. Basierend auf dem obigen Formalismus ergänzte Rechenberg die μ/λ -Notation um eckige Klammern. Die bekannten runden Klammern behalten somit ihre bisherige Bedeutung, da sie weiterhin in der Notation vorkommen und von den eckigen Populationsklammern umschlossen werden. So steht zum Beispiel folgender Ausdruck

$[3,5(4,8)]$ -ES

für drei gleichzeitig existierende Populationen, die jeweils fünf neue Populationen zeugen. Aus diesen fünf Populationen werden dann analog zum Vorgang bei den Individuen die drei Besten ausgewählt und bilden die neuen drei unabhängigen Populationen. Die Zahlen in den runden Klammern haben die gleiche Bedeutung wie zuvor. Sie verraten, dass eine Population jeweils aus vier Individuen besteht, welche acht Nachkommen erzeugen. Aus diesen acht werden dann wieder die besten vier selektiert und bilden die neue Population. Dazu bedarf es eines Qualitätsmaßes. Die Populationsqualität kann auf verschiedene Weisen bestimmt werden. Eine Möglichkeit stellt die mittlere Qualität aller Individuen einer Population dar. Eine weitere Möglichkeit ist die Repräsentation der Populationsqualität durch die Qualität des besten Individuums.

Wie auch bei den vorherigen ES kann bei den Populations-ES zwischen der „“ und „+“ Notation entschieden werden. Von der $[3,5(4,8)]$ -ES über $[3+5(4+8)]$ -ES und $[3,5(4+8)]$ -ES bis hin zur $[3+5(4,8)]$ -ES Variante ist alles möglich. Sogar die in Zusammenhang mit der sexuellen Rekombination eingeführte „/“ Notation ist hier umsetzbar. So zum Beispiel

$[6/2\#3(4\#7)]$ -ES

Die Bedeutung dieser Notation ist aus der Notation für einzelne Individuen übertragbar. Hier existieren 6 unabhängige Populationen, die sich immer zu Gruppen à zwei Populationen zusammen schließen. Aus diesen Zweiergruppen entstehen dann jeweils drei neue Populationen.

Also drei mal drei neue Populationen. Die sechs Besten dieser neun Nachfolgepopulationen, bilden dann wieder sechs unabhängige Populationen. Jede dieser sechs Populationen besteht aus vier Individuen. Diese vier Individuen zeugen jeweils sieben Kinder von denen die vier besten überleben.

Da in der Beispiel Notation das #-Symbol verwendet wird, kann wahlweise ein „+“ bzw. ein „/“ eingesetzt werden. Mit dieser Variation soll der Genfluss zwischen Populationen simuliert werden. Es wird also eine Zufällige Migration nachgestellt. Aber

auch eine geplantes Wanderverhalten ist möglich. Es besteht die Möglichkeit eine festgelegte Anzahl der Besten aus einer Population abwandern zu lassen.

Doch die Natur ist so vielfältig, dass der Erweiterung dieses Formalismus und seiner Fülle an Varianten keine Grenzen gesetzt sind. Um in dieser Arbeit noch auf eine letzte Modifikation des Formalismus einzugehen, sei ein weiterer Blick in die Natur gewagt.

Durch geographische Veränderungen kann es im Laufe von mehreren Jahrzehnten durchaus vorkommen, dass ganze Populationen für mehrere Generationen isoliert werden. Ein sehr bekanntes Beispiel ist die Insel Galapagos, auf der sich durch ihre Isolation Populationen über Generationen nahezu unbeeinflusst entwickeln konnten.

Die Simulation einer isolierten Populationen wird ebenfalls von Rechenberg mit einer weiteren Notation eingeführt. Rechenberg ergänzt eine hochgestellte Isolationszahl. Durch diese Zahl wird die Dauer einer Isolation angegeben. Ob dies nun in Sekunden, in Generationen oder einem anderen Zeitmaß stattfindet, ist dem Anwender überlassen. Ein Beispiel könnte so aussehen:

$[1,3(5,7)^{30}]$ -ES

Diese ES Variante steht für eine Population. Diese Population zeugt drei Nachfolgepopulationen. Durch den Schrägstrich wird die Isolationszahl eingeleitet. In diesem Fall soll die Zeitmessung durch Generationenanzahl erfolgen. Die Isolationsdauer von 30 Generationen besagt nun, dass die drei neu gezeugten Populationen sich 30 Generationen lang isoliert entwickeln können, bevor aus ihnen die Beste ausgewählt wird, um zu überleben.

Die überlebende Population beginnt dann wieder eine neue Evolutionskette.

Dieses Verfahren eignet sich ebenfalls für den Einsatz auf Parallelrechnern. So können durch die verschiedenen Populationen unterschiedliche Regionen im Suchraum abgesucht werden. Da in den Populationen die Individuen weiterhin optimiert werden, wird die Parallelität doppelt ausgenutzt.

3.5. Richtwerte für die Evolutionsstrategien

Durch die verschiedenen eingeführten ES ist eine unendliche Anzahl von Varianten möglich. Die kennengelernten ES für Individuen und Populationen lassen sich mit Rechenbergs Ansatz noch beliebig erweitern. Populationen von Populationen und so weiter sind umsetzbar. Der Verschachtelungen sind keine Grenzen gesetzt

$\{y'/x' \# z' [y'/x' \# z' (y/x \# z)^{1/m}]^{1/n}\} - \text{ES}$

Ob es in praktischen Anwendungen sinnvoll ist, solche Schachtelungen oder gar tiefere anzuwenden ist fragwürdig. In der Regel endet die Evolutionsstrategie mit der zweiten Iteration und besitzt somit die Form:

$[u/v \# w (x/y \# z)^n] - \text{ES}$

Bei der Betrachtung dieser Form fällt schnell auf, dass die Kombination der zu wählenden Parameter unzählig hoch ist. Es stellt sich schnell die Frage, ob es Richtwerte oder sogar optimale Parameterwerte für die Variablen u,v,w etc. gibt.

Das Verhältnis von μ und λ , also der Selektionsdruck, ist ausschlaggebend, falls ein hoher Zuverlässigkeitsanspruch an die Algorithmen gestellt wird. Von Rechenberg sind Werte von 1/5 bis 1/3 für den Quotienten μ/λ empfohlen. Daraus ist eine Tabelle ableitbar, in der die optimalen Werte für λ abhängig von μ nachzulesen sind:

μ	1	2	3	4	5	6	7	8	9	10
λ	5	8	11	13	16	19	22	25	27	30

[Rechenberg 1992]

Für die Gruppenstärke ρ , wie bei einer $(\mu/\rho\#\lambda)$ -ES verwendet wird, empfiehlt Rechenberg einen Maximalwert von $\rho = \mu$.

Die hier angegebenen „optimalen“ Parameter für eine ES sind nur Richtwerte. Die Wahl der Parameter und somit auch die optimalen Parameter einer ES sind sehr stark von dem gegebenen Problem abhängig. Die angegebenen Parameter sind bei „einfachen“ Problemen durchaus gute Richtwerte. Es gibt viele reale und komplexe Qualitäts- und Testfunktionen, bei denen diese Richtwerte absolut nicht sinnvoll sind.

3.6. Nachteile der Rechenberg-Schwefel Notation

Trotz der vielfältigen Möglichkeiten, die durch die Rechenberg-Schwefel Notation geboten werden, gibt es doch einige Schwächen, auf die im folgenden eingegangen werden soll.

Es ist mit der kennengelernten Notation nicht auszudrücken, welches Mutations- oder Rekombinationsverfahren verwendet wird. Diese Verfahren müssen gesondert beschrieben werden. Bei der $(\mu/\rho\#\lambda)$ -ES kann es von ausschlaggebender Bedeutung sein, welche Art der Rekombination genutzt wird.

Des weiteren ist es zum Beispiel in der $[u\#v(w\#x)^n]$ -ES nicht möglich, sich die v isolierten Populationen zwar gleichzeitig aber unterschiedlich entwickeln zu lassen. Es kann durchaus sinnvoll sein die eine Population n Reproduktionszyklen und eine andere m Reproduktionszyklen durchlaufen zu lassen. So könnten auch eine verschiedene Individuenanzahl und verschiedene Rekombinationsverfahren von Interesse sein. Diese vorgestellten Variationen sind mit der Rechenberg-Schwefel Notation nicht umsetzbar.

3.7. Die Mutation

Der Grundgedanke der ES ist die Nachkommenerzeugung durch DNS-Selbstverdopplung. Das Ausgangsindividuum dupliziert sich und das Duplikat wird mutiert. Diese Mutation hat in der ES eine besondere Bedeutung. Im Gegensatz zu den Verfechtern Genetischer Algorithmen, legen die Evolutionsstrategen weniger Wert auf ausgefallene Rekombinations- oder Selektionsverfahren.

Sowohl für Individuen als auch für Populationen erfolgt die Selektion, Mischung und Rekombination der Vektoren nach einer statistischen Gleichverteilung. Es findet jede Entscheidung mit der gleichen Wahrscheinlichkeit statt.

Die Selektion der Individuen zum Zwecke der Erzeugung von Nachkommen erfolgt damit unabhängig von der individuellen Fitness. [Schöneburg 1994]

Die Mutation erfolgt bei den ES nach dem Prinzip der statistischen Normalverteilung. Zur Erzeugung von Kindern wird auf den Ausgangsvektor, also den Elter, ein Zufallsvektor dazu addiert. Eine formale Beschreibung der Zeugung sieht so aus:

$$X_{\text{neu}} = X_{\text{alt}} + N(0, \sigma)$$

$N(0, \sigma)$ steht hier für einen Vektor mit Zufallszahlen als Parameter. Die Zufallszahlen sind Gauß-verteilt mit dem Mittelwert 0 und der Standardabweichung σ .

Es werden also auf die Eltervektorparameter Zufallszahlen addiert. Auf Grund des Mittelwertes schwanken die Zufallszahlen um Null. Die Variable σ steht in diesem Fall für die Schwankungsbreite der Zufallszahlen. Diese Standardabweichung kann für jede Vektorkomponente unterschiedlich sein. Es können also für die erste Komponente Zufallszahlen mit Standardabweichung 0,25, für die zweite 0,666 und für die dritte 1 erzeugt werden. Die Wahl des Mittelwertes Null führt zu einer Bevorzugung von kleinen Veränderungen.

Zur Verdeutlichung dient ein weiteres Beispiel:

Der Ursprungsvektor sei gegeben durch:

$$X_{\text{alt}} = \langle -0.75, 0.38, 3 \rangle$$

die Standardabweichung für die einzelnen Komponenten, sei wie folgt gewählt:

$$\sigma = \langle 1, 0.5, 6 \rangle$$

Die damit erzeugten Gauß-Verteilten Zufallszahlen seien:

$$N(0,1) = -0.8$$

$$N(0,0.5) = 0.1$$

$$N(0,6) = -3.2$$

Daraus ergibt sich der Nachkomme:

$$X_{\text{neu}} = X_{\text{alt}} + N(0,\sigma) = \langle -1.55, 0.48, -0.2 \rangle.$$

Bei Betrachtung der Eltern als Punkte im n-dimensionalen Raum, erzeugt das beschriebene Mutationsverfahren eine potentielle Sphäre von Nachkommen um jeden Elter. Somit befinden sich in jeder Raumrichtung potentiell gleich viel Nachkommen. Nah am Sphärenmittelpunkt gibt es sehr viele mögliche Nachkommen, während mit wachsendem radialen Abstand es immer weniger werden.

4. Anwendung auf das Travelling-Salesman-Problem

Bei dem Travelling-Salesman-Problem (TSP) ist eine definierte Menge von Städten und Entfernungen zwischen diesen Städten gegeben. Nun soll eine Route durch diese Städte ermittelt werden, die jede Stadt genau einmal durchquert, am Ende wieder an dem Startpunkt endet und von minimaler Distanz ist. Diese Route wird dann als optimale Lösung angesehen. Bei der Suche nach dieser Route wird in den meisten Lösungsansätzen in jeder Stadt entschieden, welche Stadt als nächstes durchquert werden soll. Dieses wäre jedoch schwer in den Genetischen Algorithmen zu modellieren, daher repräsentiert ein Individuum in einer Population eine mögliche Route, die jede Stadt genau einmal durchquert. Dieser Ansatz kann verglichen werden mit einer Gruppe von „blinden Handelsreisenden“, die eine zufällige Route wählen und erst am Ende Aussagen über die von ihnen gefahrene Route treffen können.

Wie schon in den vorherigen Kapiteln erwähnt muss nun eine geeignete Lösungsrepräsentation gefunden werden und die Operatoren muss so modifiziert werden, das sie unserer Problemstellung entsprechend gültige Lösungen erzeugen.

Der erste Ansatz für eine geeigneter Repräsentation könnte anhand eines binären Vektors erfolgen. Angenommen wir haben die vier Städte A,B,C,D gegeben, dann könnte jede Stelle in dem Vektor für eine Kante zwischen zwei Städten stehen. Dann würde die Route A → B → C → D die Kanten ab, bc, cd und da enthalten. Alle möglichen Kanten zwischen den Städten sind die Kanten ab, ac, ad, bc, bd, cd. Hierbei ist zu beachten das diese Kanten ungerichtet sind und somit die Kante „ab“ der Kante „ba“ entspricht. Ausgehend von der gerade angesprochenen Reihenfolge der Kanten könnte der Vektor für unsere Route folgendermaßen aussehen:

ab	ac	ad	bc	bd	cd
1	0	1	1	0	1

Dieses stellt sicherlich eine mögliche Lösungsrepräsentation dar, jedoch können Schwierigkeiten auftreten, wenn wir mit Hilfe dieser Repräsentation neue Individuen durch Rekombination und Mutation erzeugen wollen, die immer noch unsere geforderten Bedingungen für eine gültige Lösung erfüllen. Denn mit unseren üblichen

Verfahren können Routen entstehen, die keine geschlossene Strecke darstellen oder die manche Städte mehrfach oder gar nicht durchqueren. Auch das Überprüfen auf die Gültigkeit einer Lösung ist hier sehr aufwendig, was zu späterem vermehrten Rechenaufwand führen wird, wodurch die Effizienz unseres Algorithmus stark vermindert wird.

Ein alternativer naheliegender Ansatz ist die Darstellung einer Lösung durch einen Vektor von Indizes, wobei jeder Indize für eine Stadt steht. Somit muss bei einer Problemstellung von N Städten jeder Vektor N Indizes enthalten und jeden Indize von 0 bis N-1 genau einmal enthalten.

Mögliche Repräsentationen bei einer Problemstellung von zehn Städten könnte dann folgendermaßen aussehen:

$L_1 = (1, 5, 9, 3, 6, 8, 0, 2, 4, 7)$

$L_2 = (0, 6, 3, 9, 5, 2, 1, 7, 8, 4)$

$L_3 = (5, 3, 7, 8, 4, 2, 0, 6, 1, 9)$

...

Operatoren, die bei solch einer Repräsentation gültige Lösungen erzeugen, wurden bereits in dem Kapitel 2.5.2 unter dem Stichwort Sequenzoperatoren vorgestellt. Bis jetzt wurde jedoch nur der Operator des Crossover auf Sequenzen angepasst. Im folgenden wird nun eine Variante des Mutationsoperators vorgestellt, der sicher stellt, dass auch mutierte Lösungen noch gültige Lösungen sind.

Es werden nun nicht einzelne Elemente der Individuen mutiert, sondern die Reihenfolge der Elemente der Sequenzen geringfügig verändert, da dieses eine Qualitätsänderung zur Folge hat. Bei einer gegebenen Sequenz werden zunächst zwei Punkte innerhalb der Sequenz ausgewählt. Das Element an der vorderen Position wird aus der Sequenz gelöscht und alle Elemente rechts von dieser Position einschließlich des Elements an der zweiten Position werden um eine Position nach links verschoben. In die nun entstandene Lücke an der hinteren Position wird nun das zuvor gelöschte Element eingefügt.

Hierdurch wurde eine gültige Lösung erzeugt, die sich geringfügig von der vorherigen Lösung unterscheidet und somit dieses Verfahren als Mutationsoperator anwendbar ist.

Nachdem eine passende Lösungsrepräsentation und Sequenzoperatoren zur Verfügung stehen, kann das TSP nun simuliert werden. Bei einer Simulation des TSP [Schöneburg 1994], wobei 25 deutsche Städte und deren Distanzen gegeben sind, wurde die optimale Lösung schon bereits nach 137 Generationen gefunden.

5. Vergleich zwischen ES und GA

Beide Verfahren beginnen mit einer breiten Verteilung der Individuen im Suchraum. Abgeschlossen wird sowohl bei den ES als auch bei den GA mit der Konzentration der Individuen auf einen minimalen Ausschnitt des Lösungsraums. In den ES wie auch in den GA wurde eine noch immer wachsende Anzahl von Varianten entwickelt. Durch diese nicht mehr überschaubare Varianten verschwimmen die Grenzen zwischen den Verfahren. Es kommt sogar vor, dass bei Genetischen Algorithmen anstelle der üblichen binären Vektoren, für bestimmte Problemstellungen reelle Vektoren eingesetzt werden. Welcher Algorithmus bei welcher Problemstellung besser ist, kann nicht eindeutig entschieden werden. Die Algorithmen arbeiten sehr abhängig von der Problemstellung. Es lassen sich nur sehr wenige allgemeine Eigenschaften der beiden Algorithmen finden. So zum Beispiel ihr Verhalten in ganz bestimmten Suchräumen. Ist ein Suchraum gegeben, dessen Extremwert, von einem sehr flachen Raum umgeben ist, so

finden beide Algorithmen nur sehr schwer eine Lösung. Das liegt daran, dass die Algorithmen keine Anhaltspunkte haben. Sie können die Population nicht gezielt verschieben. Hier versagen allerdings auch die meisten klassischen Verfahren. Den einzigen Vorteil den sowohl die ES als auch die GA haben, ist die Möglichkeit eines zufälligen Findens.

Oft ist in Quellen von einer strikten Trennung zwischen ES und GA zu lesen. Da sich aber beide Verfahren noch immer weiter entwickeln und an Verfeinerung dazu gewinnen, ist die Trennung kaum noch gültig. Bei Verwendung elementarer ES oder GA nach dem Vorbild der ersten entwickelten Verfahren, ist eine solche Argumentation denkbar. Aus diesem Grund sei darauf hingewiesen, dass der folgende Vergleich in dieser Form nicht als allgemeingültig gelten kann:

Durch die starke Mutation und Selektion bei den ES, werden bei diesem Verfahren viel schneller homogene Populationen erreicht als bei GA. Das entspricht einem schnelleren Auffinden lokaler Extremwerte. Für Problemstellungen, bei denen nicht unbedingt die beste Lösung gefunden werden muss, ist deshalb das Anwenden von ES ratsamer. Sie finden schneller beliebige, gute Lösungen und „vernachlässigen“ die Suche nach dem absoluten Extremwert.

Da bei den GA sowohl die Mutation als auch die Selektion sehr schwach gewählt werden, wird der Suchraum viel breiter durchsucht. Hier ist die schnelle Homogenisierung von Populationen gezielt unterdrückt.

Wie schon erwähnt, kann diese Art des Vergleiches zu voreiligen und falschen Schlussfolgerungen führen. Die Effizienz der beiden Verfahren hängt von vielen Faktoren ab. Die Parametereinstellungen, die Problemstellung und nicht zu letzt die Codierung eines Problems tragen entscheidend zu der Effektivität der Verfahren bei.

6. Literaturverzeichnis

- | | |
|--------------------|--|
| [Darwin 1859] | Darwin, Charles, <i>On the Origin of Species by Means of natural Selection</i> . Murray London 1859 |
| [Heistermann 1994] | Heistermann, Jochen, <i>Genetische Algorithmen</i> . Teuber Stuttgart[u.a.] 1994 |
| [Rechenberg 1973] | Rechenberg, Ingo, <i>Evolutionsstrategie</i> . Friedrich Frommann Verlag Stuttgart, 1973 |
| [Rechenberg 1992] | Rechenberg, I., <i>Evolutionsstrategie '92</i> . unveröffentlichtes Manuskript, TU Berlin, 1992 |
| [Nissen 1997] | Nissen, Volker, <i>Einführung in Evolutionäre Algorithmen</i> . Braunschweig [u.a.] : Vieweg , 1997 |
| [Schöneburg 1994] | Schöneburg, Eberhard, <i>Genetische Algorithmen und Evolutionsstrategien</i> . Bonn; Paris; Reading Mass. [u.a.] : Addison-Wesley 1994 |