

Einführung genetischer Algorithmen mit Anwendungsbeispiel

Steffen Harbich

26. Dezember 2007

Zusammenfassung Im Rahmen des Seminars „Das virtuelle Labor“ entstand die vorliegende Ausarbeitung zum Thema „genetische Algorithmen“. In Analogie zur biologischen Evolution gelangen genetische Algorithmen ausgehend von einer zufällig gewählten Menge von Anfangslösungen eines Problems durch das wiederholte Verändern, Kombinieren und Auswählen von Lösungen zu einem ausreichend gutem Ergebnis. Aufgrund ihrer Einfachheit eignen sich genetische Algorithmen vor allem dann, wenn über die Struktur des zu lösenden Problems wenig bekannt oder aber wenn die Menge möglicher Lösungen sehr groß ist.

Anhand der Routenplanung eines Paketdienstes als begleitendes Anwendungsbeispiel aus dem Alltag wird der Aufbau und die Funktionsweise genetischer Algorithmen eingeführt. Die biologischen Grundlagen zur Evolution sowie benötigtes Wissen zur Problemklasse der Optimierungsprobleme werden zuvor vermittelt. Abschließend werden weitere Anwendungsbeispiele und Vor- und Nachteile genetischer Algorithmen beschrieben.

Inhaltsverzeichnis

1	Einleitung	3
2	Grundlagen	3
2.1	Optimierungsprobleme	3
2.2	Biologische Evolution	4
3	Genetische Algorithmen	5
3.1	Definition	6
3.2	Startpopulation	6
3.3	Fitnessfunktion	6
3.4	Selektion	7
3.5	Genetische Operatoren	7
3.6	Abbruchkriterium	8
4	Anwendungsbeispiele	9
5	Fazit	10
	Literatur	11

1 Einleitung

Stellen Sie sich vor, Sie fliegen während des Urlaubs in ein Ausland ihrer Wahl und haben sich vorgenommen eine handvoll Städte zu besichtigen. Damit Sie nicht zu viel Zeit verbringen, in einem Fahrzeug von Stadt zu Stadt zu fahren und ihren Urlaub besser genießen können, haben Sie sich gefragt, welche Reihenfolge der Städte ihnen die geringste Fahrzeit beschert. Nun, die Distanzen in Kilometern zwischen den Städten können von einem Routenplaner berechnet werden. Da Sie von einer konstanten Geschwindigkeit beim Fahren ausgehen, ist die Entfernung ein Indikator für die Zeit, die im Fahrzeug benötigt wird. Wie finden Sie jetzt die schnellste Route, bei der alle Städte genau einmal besucht werden?

Dieses Problem ist bekannt als das Problem des Handlungsreisenden (*TSP*, engl. Travelling Salesman Problem). Eine optimale Lösung des Problems erhält man beispielsweise, indem alle Möglichkeiten einer Rundreise durch die Städte aufgestellt werden und die kürzeste ausgewählt wird. Jedoch ist der Aufwand eines Algorithmus für diesen Lösungsweg sehr hoch, wie man sich leicht vor Augen führt. Denn bei n Städten ergeben sich insgesamt $n \cdot (n - 1) \cdot \dots \cdot 2 \cdot 1 = n!$ Möglichkeiten, die verarbeitet werden müssen. Wollten Sie zehn Städte im Urlaub besichtigen, so gäbe es bereits $10! = 3.628.800$ mögliche Rundreisen. Dieses Beispiel mit einem Computer zu berechnen ist heutzutage kein Problem. Aber stellen Sie sich nun vor, ein Paketdienst möchte 30 Pakete in einer Stadt ausliefern. Dies ergibt mehr als $2,6 \cdot 10^{32}$ Möglichkeiten! Das TSP gehört zu der Klasse von Problemen, die vermutlich nicht mit polynomialem Zeitaufwand optimal gelöst werden können [WikiTSP].

Um eine Lösung in vertretbarem Zeitaufwand zu erlangen, benötigen wir demnach andere Methoden und Verfahren. Im weiteren Verlauf der Arbeit werden wir sehen, dass genetische Algorithmen (*GA*) geeignet sind, um Probleme wie das TSP effizient zu lösen. Doch zunächst soll das folgende Kapitel die Grundlagen für das Beschreiben und Aufstellen von genetischen Algorithmen legen.

2 Grundlagen

2.1 Optimierungsprobleme

Genetische Algorithmen sind Optimierungsverfahren, welche Optimierungsprobleme lösen [WikiGA]. Es folgt daher eine Definition der Problemklasse.

Bei einem Optimierungsproblem sind ein Lösungsraum S (Menge von möglichen Lösungen, auch *Suchraum* genannt) und eine *Zielfunktion* $f : S \rightarrow \mathbb{R}$ gegeben. Die Zielfunktion (auch *Bewertungs-* oder *Fitnessfunktion* genannt) ordnet jeder Lösung $s \in S$ die Güte der Lösung s zu. Gute Lösungen haben dabei einen größeren Wert als schlechtere Lösungen. Gesucht ist nun eine Lösung $s \in S$, sodass $f(s) = \max \{f(x) \mid x \in S\}$. Die Optimierung besteht demnach darin, die Zielfunktion zu maximieren.

Diese Form von Optimierungsproblemen wird deshalb als Maximierungsproblem bezeichnet. Soll die Zielfunktion minimiert werden, liegt ein Minimierungsproblem vor, dass

sich auf ein Maximierungsproblem mit Zielfunktion $\tilde{f} : S \rightarrow \mathbb{R}$, $\tilde{f}(x) = -f(x)$ zurückführen lässt.

In vielen Fällen liegt mehr als nur ein Kriterium vor, nach denen optimiert werden soll. Allgemein gibt es dann k Optimierungskriterien und $k \in \mathbb{N}$ zugehörige Zielfunktionen $f_i : S \rightarrow \mathbb{R}$ ($1 \leq i \leq k$). Ein Ansatz zur Lösung dieser Mehrkriterienoptimierungsprobleme besteht darin, die Zielfunktionen zu einer Zielfunktion $g : S \rightarrow \mathbb{R}$ mit $g(x) = \sum_{i=1}^k g_i \cdot f_i(x)$ zusammenzufassen und diese zu maximieren. Die Variable $g_i \in \mathbb{R}$ sind hierbei festzulegende Gewichte der einzelnen Bewertungsfunktionen [Evo04, S.7-9].

Das Beispiel aus der Einleitung können wir nun wie folgt beschreiben. Sei $M = \{s_1, \dots, s_n\}$ die Menge der Städte und $d : M \times M \rightarrow \mathbb{R}$ eine Funktion, die die Entfernungen zwischen den Städten modelliert, wobei von jeder Stadt ein Weg zu jeder anderen Stadt existiere. Dann lässt sich der Suchraum und die Zielfunktion angeben als

$$S = \{(s'_1, s'_2, \dots, s'_n) \mid s'_i \in M, s'_i \neq s'_k, 1 \leq i, k \leq n, k \neq i\}$$

$$f(s'_1, s'_2, \dots, s'_n) = \sum_{i=1}^{n-1} d(s'_i, s'_{i+1}).$$

Die Bewertungsfunktion berechnet also die Gesamtlänge der Strecke, die bei der Rundreise zurückgelegt wird. Ziel ist es, diese zu minimieren.

Wie beim TSP ist die Bestimmung einer optimalen Lösung bei vielen Optimierungsproblemen nicht in annehmbarer Zeit zu bewerkstelligen. Meist genügt deshalb eine suboptimale Lösung, das heißt ein lokales Optimum, das sich dem globalen Optimum ausreichend nähert [Evo04, S.13].

Neben den genetischen Algorithmen gibt es eine Vielzahl anderer Optimierungsverfahren. Beipielsweise können lineare Optimierungsprobleme mit den Werkzeugen der Mathematik auf analytischem Wege berechnet werden.

Als nächstes schauen wir uns die biologischen Grundlagen an, die das Verständnis der Vorgehensweise von genetischen Algorithmen vereinfacht.

2.2 Biologische Evolution

Genetische Algorithmen sind an die biologische Evolution und an die Genetik angelehnt und abstrahieren deren Terminologie. Es soll daher kurz auf die wichtigsten Begriffe und die Funktionsweise der Evolution eingegangen werden.

Die Grundbausteine jedes Lebewesens sind Zellen. Die überwiegende Mehrheit vorkommender Zellen bilden die eukaryotischen Zellen. Sie enthalten einen Zellkern, in dem sich die *Chromosomen* befinden, die Träger der Erbsubstanz. Das Erbgut – kodiert in der Erbsubstanz – ist der „Bauplan“ für das Lebewesen (*Individuum*). Abbildung 1 zeigt den schematischen Aufbau eines Chromosoms. Chromosome lassen sich in kleine Teilstücke, die sogenannten *Gene*, unterteilen, die eine bestimmte Teileigenschaft wie beispielsweise die Augenfarbe festlegen. Gene können nicht beliebige Zustände

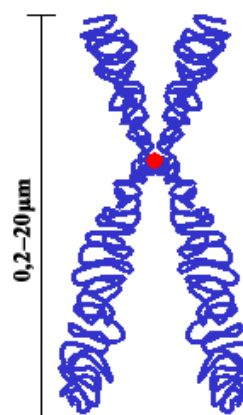


Abbildung 1: Schema eines Chromosoms [WikiChr]

annehmen, sondern besitzen festgelegte Ausprägungen, welche als *Allele* bezeichnet werden. Es gibt zum Beispiel Allele für braune, grüne und blaue Augenfarben, jedoch keines für gelbe oder pinke. Meist wirken mehrere Gene bei der Ausprägung eines Merkmals zusammen wie etwa bei der Hautfarbe des Menschen [Evo04, Anhang A].

Evolution ist die Veränderung der vererbaren Merkmale – sprich der Gene – einer Population¹ von Lebewesen von Generation zu Generation. Die Gene werden bei der Fortpflanzung an den Nachwuchs weitergegeben [WikiEvo]. Dabei gibt es drei grundlegende Mechanismen, die die Evolution prägen:

- **Selektion** ist die natürliche Auslese von Individuen. Lebewesen, die an ihre Umwelt oder deren Veränderung gut angepasst sind, haben eine hohe Überlebenswahrscheinlichkeit. Dies hat zur Folge, dass sie sich mit erhöhter Wahrscheinlichkeit fortpflanzen und somit ihre angepassten Gene weitergeben können. Die Anpasstheit eines Individuums wird als *Fitness* bezeichnet.
- Durch **Mutation** verändern sich die Allele von Genen. Ursache dafür können physikalischer oder chemischer Natur sein wie zum Beispiel Radioaktivität oder das Vorhandensein von Mutagenen². Mutationen können Lebewesen sowohl schädigen als auch überlebenstüchtiger werden lassen.
- Bei der **Rekombination** (*Crossover*) entstehen neue Gen- und Merkmalskombinationen durch Austausch von Allelen zweier Chromosomen. Es findet also ein Mischung von Merkmalen statt. Die Rekombination und die Mutation zusammen sind verantwortlich für die genetische Variabilität einer Population.

Über Tausende von Generationen sorgt das Wechselspiel dieser Mechanismen für die Anpassung der Population an die Umwelt- und Lebensbedingungen. Die Evolution löst damit ein schwieriges Optimierungsproblem, bei dem die Überlebenswahrscheinlichkeit der Individuen zu maximieren ist.

Wir haben nun die nötigen Grundlagen, um uns im nächsten Kapitel den genetischen Algorithmen zuzuwenden.

3 Genetische Algorithmen

Aus dem Kapitel *Grundlagen* wissen wir bereits, dass das Problemfeld von GAs Optimierungsprobleme sind und die Evolution so interpretiert werden kann, dass sie ein schwieriges Optimierungsproblem löst. Es liegt daher nahe die „Strategie“ der Evolution zur Lösung solcher Probleme anzuwenden.

Die grundlegende Idee dabei ist, anfängliche „Lösungsvorschläge“ solange zu verändern und zu kombinieren bis eine ausreichend gute Lösung für das Problem gefunden wurde. Dies führt zu folgender Definition von GAs.

¹Unter einer Population versteht man hier die Gesamtheit der Individuen einer Art.

²Mutagene sind Mutationsauslösende Stoffe.

3.1 Definition

Ein genetischer Algorithmus ist ein Algorithmus der Form:

Gegeben: Optimierungsproblem mit Suchraum S und Fitnessfunktion f

Lösung:

```
Generiere zufällige Startpopulation P
SOLANGE (Abbruchbedingung nicht erfüllt)
{
    Berechne Fitness der Individuen aus P
    Selektiere nach Fitness
    Wende genetische Operatoren an
}
```

Ergebnis: bestes Individuum aus der Population

Ein „Individuum“ der Population P wird als Zeichenkette, Zahlenfolge oder Folge von Variablen binär kodiert und man spricht statt vom Individuum auch nur vom Chromosom, das repräsentativ für das „Lebewesen“ ist. In Analogie zur Genetik sind Gene die einzelnen Zeichen oder Variable und Allele die Werte, die diese annehmen können. Ein Individuum beziehungsweise Chromosom entspricht einer Lösung aus dem Suchraum S des Problems.

3.2 Startpopulation

Der erste Schritt des Algorithmus besteht darin, die erste Generation der Population (Startpopulation) mit zufälligen Merkmalen zu erzeugen. Üblicherweise werden hierfür zufällig Chromosomen aus dem Suchraum S gewählt. Prinzipiell ist es auch möglich, die Startpopulation mit gemäß der Fitnessfunktion „guten“ Chromosomen zu belegen. Dabei geht man jedoch das Risiko ein, dass diese sich durchsetzen, obwohl sie zu einem lokalen Optimum und damit zu einer suboptimalen Lösung führen. Dieser Effekt wird als *vorzeitige Konvergenz* bezeichnet.

3.3 Fitnessfunktion

Im nächsten Schritt wird mit Hilfe der Fitnessfunktion jedem Chromosom c aus $P = \{c_1, c_2, \dots, c_n\}$ die Fitness $f(c)$ zugeordnet. Die Fitnessfunktion entspricht der Zielfunktion des Optimierungsproblems. Aus den Fitnesswerten können die Überlebenswahrscheinlichkeiten $p(c_i)$ der Chromosomen $c_i \in P$ nach der Formel

$$p(c_i) = \frac{f(c_i)}{\sum_{j=1}^n f(c_j)}$$

berechnet werden. Diese werden für die Selektion benötigt. Je größer $f(c)$ gegenüber der Fitness anderer Chromsomen ist, desto größer ist auch $p(c)$. Der Leser möge sich bitte vor Augen führen, dass $\sum_{c \in P} p(c) = 1$. Für den Sonderfall, dass $\sum_{j=1}^n f(c_j) = 0$, so sind $p(c_i) = 1/n$ zu setzen.

3.4 Selektion

Als nächstes wählt der Algorithmus zufällig diejenigen Chromosomen aus, die die nächste Generation der Population bilden. Dabei spielt die Überlebenswahrscheinlichkeit $p(c)$ des Chromosoms c eine wesentliche Rolle. Sie gibt die Wahrscheinlichkeit an, mit der ein Chromosom ausgewählt wird. Algorithmisch wird dies wie folgt umgesetzt:

Auf einem Intervall $[0, 1]$ werden die Überlebenswahrscheinlichkeiten der Chromosomen nacheinander markiert. In Abbildung 2 ist diese Markierung für die Beispielwerte $p(c_1) = 0,35$, $p(c_2) = 0,1$, $p(c_3) = 0,1$ und $p(c_4) = 0,45$ dargestellt. Eine anschlie-

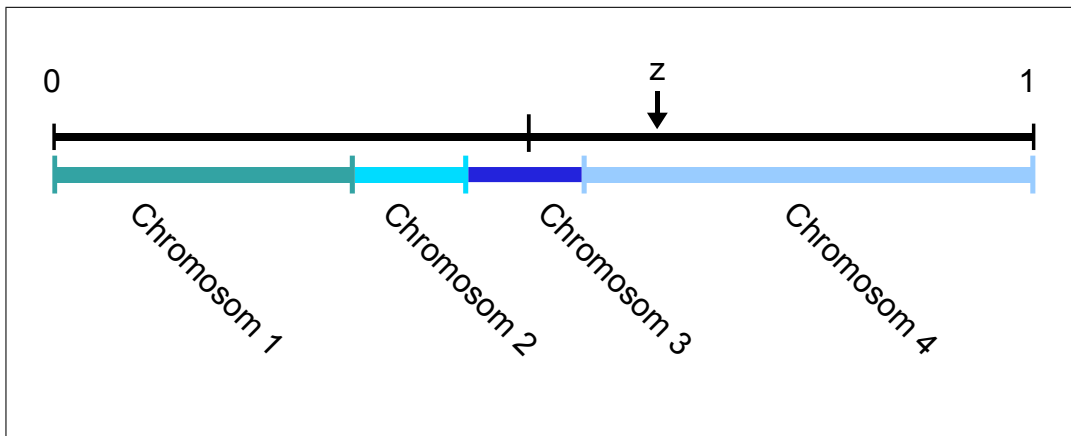


Abbildung 2: Intervalleinteilung für Selektion von Chromosomen

ßend generierte, reelle Pseudozufallszahl z zwischen 0 und 1 zeigt wie in Abbildung 2 auf ein Chromosom, welches für die nächste Generation ausgewählt wird. Der Vorgang wird genau n -mal wiederholt, wobei mehrfach ausgewählte Chromsomen eindeutig umbenannt werden. Die neue Generation enthält dann genauso viele Chromosomen wie die Vorgängergeneration. Auf die neue werden als nächstes die genetischen Operatoren angewendet.

3.5 Genetische Operatoren

Nachdem die Selektion für die Auswahl der „besten“ Chromosomen (engl. *survival of the fittest*) gesorgt hat, liegt es nun an der *Mutation* und am *Crossover*-Operator, die Chromosomen zu verändern, um neue und insbesondere fittere Chromosomen zu erlangen.

Die Mutation verändert zufällig ein oder mehrere Gene eines Chromsoms. Für jedes Gen jedes Chromsoms aus P wird mit Hilfe einer gegebenen Mutationswahrscheinlichkeit p_m bestimmt, ob dieses mutiert oder nicht. p_m wird dabei meist auf $1/m$ gesetzt, wobei m die Anzahl der Gene pro Chromosom ist. Für den Fall der Mutation wird das mutierende Gen durch ein zufällig gewähltes, anderes Allel ersetzt. Abbildung 3a stellt den Vorgang schematisch dar.

Beim Crossover werden aus der Population P paarweise Chromosomen ausgewählt, die dann mit einer Wahrscheinlichkeit p_c zu kreuzen sind. Dabei wird jedes Chromosom genau einmal involviert. Der Crossover-Operator lässt aus zwei Chromosomen $c_1, c_2 \in$

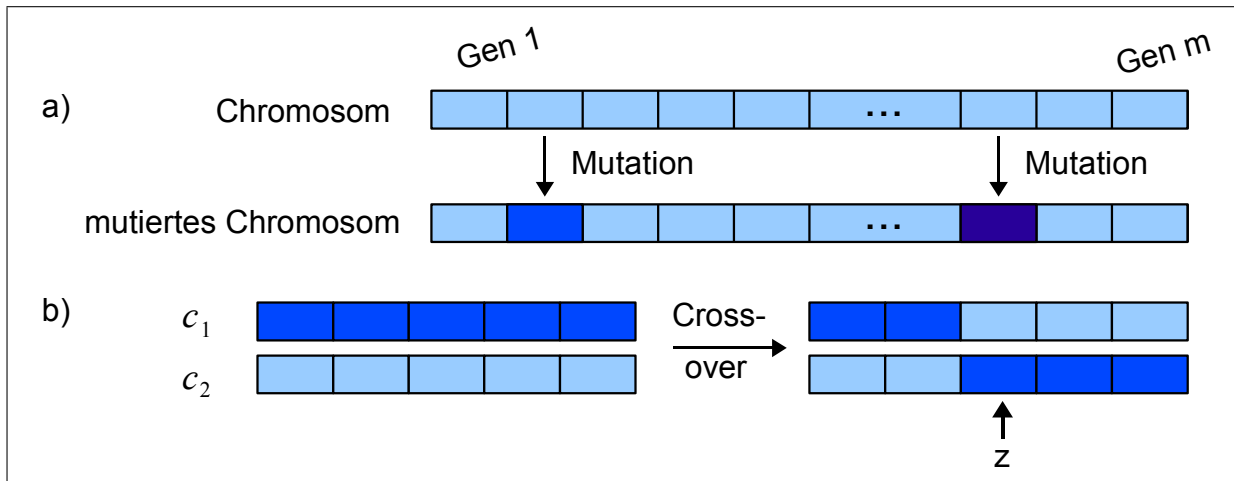


Abbildung 3: a) Mutation und b) Crossover von Chromsomen

P mit $c_1 \neq c_2$ durch Austausch von Teilen (z.B. Teile von Zeichenketten) zwei neue Chromosomen entstehen. Dazu wird eine natürliche Pseudozufallszahl z zwischen 1 und m generiert, die die Position des Austausches darstellt. Alle Gene, die in c_1 bzw. c_2 ab der Position z vorkommen, werden mit den entsprechenden Genen von c_2 bzw. c_1 getauscht. Abbildung 3b stellt den Vorgang für das Beispiel $z = 3$ schematisch dar.

3.6 Abbruchkriterium

Die drei Schritte Fitnessberechnung, Selektion und Anwendung genetischer Operatoren können beliebig oft wiederholt werden. Mit jeder neuen Generation können Änderungen auftreten, die neue und bessere Lösungen in Form von Chromosomen hervorbringen. Leider ist nicht garantiert, dass die neuen Lösungen immer besser sind als die vorherigen. So könnte sich etwa nach einer Weile eine Anzahl von Chromosomen „durchgesetzt“ haben, sodass keine wesentliche Änderung mehr auftritt. Möglich wäre auch, dass sich die Fitness der Chromosomen durch Mutation oder Crossover verschlechtert. Oder aber das Optimum wurde gefunden und ermöglicht deshalb keine weitere Verbesserung der Lösung.

Es ist daher sinnvoll und notwendig ein Abbruchkriterium zu definieren, dass dem Algorithmus mitteilt, wenn die Chromosomen der Population eine genügend gute Lösung repräsentieren. Einige Beispiele für Abbruchkriterien sind:

- Der Algorithmus hat bestimmte Anzahl von Generationen durchlaufen.
- Es ist keine Verbesserung zur vorherigen Generation aufgetreten.
- Die mittlere Verbesserung der letzten Generationen hat einen Wert unterschritten.

Mit der Erfüllung des Abbruchkriteriums endet der Algorithmus und gibt als Ergebnis das Chromosom zurück, welches die größte Fitness der Population besitzt.

Wir haben nun gesehen, wie genetische Algorithmen arbeiten und wie sie von einem Optimierungsproblem zu einer guten bis optimalen Lösung gelangen. Umfassendere Ausführungen sind in [Evo04, Kapitel 4] nachzulesen. Als Nächstes sollen ein paar Anwendungsbeispiele aus der Praxis die Bedeutung von GAs verdeutlichen.

4 Anwendungsbeispiele

Es gibt heute eine Vielzahl von Gebieten, in denen die genetischen Algorithmen zur Problemlösung eingesetzt werden. Das Profil eines Flügels und die Form des Rumpfes von Flugzeugen werden beispielsweise durch GAs optimiert und per Computersimulation getestet. Ebenso lassen sich beim Brückenbau die Lage, Form und das Gewicht einzelner Brückenbestandteile durch den Einsatz von GAs verbessern. Auch werden sie zum Einstellen des Schwellenpotentialvektors sowie der Netztopologie von neuronalen Netzen verwendet. Desweiteren unterstützen sie die Fahr-, Stunden- und Raumplanerstellung, angewandt beispielsweise in öffentlichen Einrichtungen wie Schulen oder Universitäten.

Allgemein helfen GAs beim Lösen NP-schwerer Aufgaben wie zum Beispiel beim Problem des Handlungsreisenden oder dem Rucksackproblem. Nähere Information dazu sind in [WikiGA] nachzulesen. Aus der Einleitung kennen wir bereits das TSP in Bezug auf das Beispiel des Paketdienstes, welches im Folgenden näher beschrieben wird.

Logistik ist die Planung, Steuerung und Kontrolle von Material-, Personen-, Energie-, Finanz- und Informationsflüssen [WikiLog]. Ein Beispiel für die Probleme, die dabei zu lösen sind, ist die Planung der Route eines Paketdienstes [Dom90]. Angenommen ein städtischer Paketdienst hat durch seine Kunden den Auftrag bekommen, Pakete aus einem Warenlager zu den Adressen (Stationen) der Kunden zu liefern (siehe Abbildung 4). Möge die Anzahl der Pakete so gering sein, dass ein einziger LKW genügt, um alle Pakete gleichzeitig transportieren zu können. Die Frage, die sich beim Ausliefern stellt, lautet: Welchen Weg hat der Fahrer des LKWs zu fahren, damit die Fahrstrecke kurz, am besten minimal bleibt?

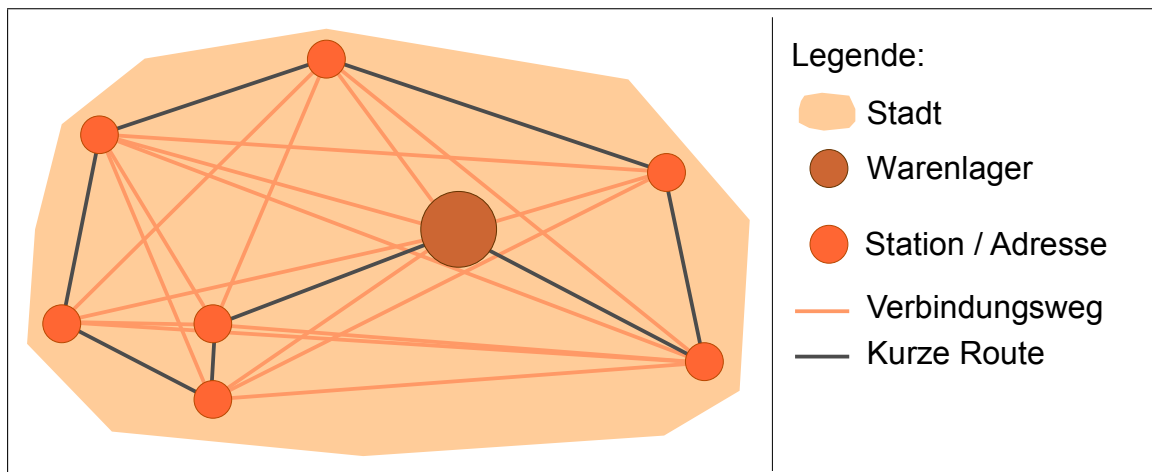


Abbildung 4: Schematisches Beispiel für die Routenplanung eines Paketdienstes

Zum Ermitteln eines genügend guten Ergebnisses stellt man einen genetischen Algorithmus auf, der die Kodierung der Lösungen (Chromosomen) als n -Tupel von Zahlen vornimmt, wobei n die Anzahl der (durchnummerierten) Stationen ist. Die Zahlen eines Tupels stellen die Reihenfolge der Auslieferung dar. Dabei ist zu beachten, dass alle Stationen je Tupel vorhanden sind. Die Fitnessfunktion ordnet den Tupeln die Länge des abzufahrenden Weges multipliziert mit -1 zu, sodass maximiert werden kann. Im Kapitel „Optimierungsprobleme“ wurde bereits eine mathematische Notation hierfür gegeben. Als Mutationsoperator eignet sich das Ändern einer Zahl eines Tupels. Allerdings muss das Tupel an einer zweiten Stelle angepasst werden, damit die Lösung ihre Gültigkeit bewahrt. Denn die geänderte Zahl, die einer gewissen Station entspricht, sollte dann wieder irgendwo im Tupel auftreten. Beim Crossover muss ebenfalls darauf geachtet werden, dass die gekreuzten Chromosomen gültige Lösungen repräsentieren. Für den Fall, dass dies nicht zutrifft, sollte das Crossover rückgängig gemacht werden. Die Mutations- und die Crossoverwahrscheinlichkeit können durch systematisches Probieren bestimmt werden. Das Chromosom mit der besten Fitness gibt am Ende des Algorithmus das Ergebnis als Reihenfolge der anzufahrenden Stationen an.

Dass genetische Algorithmen in der Wirtschaft tatsächlich eingesetzt werden, zeigt die Firma DNA-Evolutions (<http://www.dna-evolutions.com>) mit ihrem Produkt *JOpt*. Die Programmbibliothek des Produkts ermöglicht die Optimierung von Touren- und Routenplanung mittels genetischem Algorithmus.

5 Fazit

Wir haben gesehen, dass genetische Algorithmen ausgehend von beliebigen Anfangslösungen durch wiederholtes Selektieren und Verändern der Lösungen zu einem Ergebnis gelangen. Dabei ergeben sich eine Reihe von Vor- und Nachteilen, die in folgenden Absätzen beschrieben werden.

Ein größerer Nachteil genetischer Algorithmen ist, dass sie nicht immer die optimale Lösung eines Problems liefern. Dies passiert, wenn die Chromosomen der Population sich einem lokalen Optimum der Zielfunktion nähern und die genetischen Operatoren nicht ausreichend starke Änderungen hervorbringen, um die suboptimale Lösung zu überwinden. Auch funktionieren GAs effizient unter der Annahme, dass ähnliche Lösungen zu einem ähnlich gutem Ergebnis führen, was nicht unbedingt gegeben sein muss. Ein weiterer Kritikpunkt von GAs ist der relativ hohe Rechenaufwand, der durch das häufige Erzeugen neuer Generationen entsteht.

Allerdings kann die Laufzeit auch viel geringer gegenüber alternativer Optimierungsverfahren ausfallen, sofern eine an das Optimum angenäherte Lösung akzeptabel ist. Dies ist insbesondere bei NP-schweren Problemen wie dem TSP der Fall, bei denen eine Bestimmung des globalen Optimums zu rechenaufwändig ist. Ein weiterer Vorteil genetischer Algorithmen birgt das einfache Konzept und der geringe Implementierungsaufwand. Wesentlich ist dabei, dass ein nur sehr geringer Formalisierungs- und Modellierungsaufwand des Problems anfällt. Außerdem können GAs ohne großes Wissen über die Struktur des Problems umgesetzt werden, da die Menge zulässiger Lösungen, die

Kodierung der Lösungen und die Berechnung der Fitnessfunktion bereits alle nötigen Informationen enthalten. Dies macht genetische Algorithmen flexibel und für eine Vielzahl von Problemen anwendbar.

Zusammenfassend kann also gesagt werden, dass GAs eine praktikable Alternative zu klassischen Lösungsstrategien sind. Sie sollten insbesondere dann verwendet werden, wenn über das Problem nicht viel strukturelles Wissen vorliegt, der Modellierungsaufwand des Problems gering sein soll oder der Suchraum des Problems sehr groß ist.

Ein paar abschließende Worte seien zur Literatur gesagt. Für alle Leser, die ein detaillierteres Wissen über genetische Algorithmen erlangen wollen, empfehle ich die Lektüre des Buches „Evolutionäre Algorithmen“ [Evo04].

Literatur

- [Dom90] Wolfgang Domschke. *Logistik Bd. 2: Rundreisen und Touren*, 3. Auflage, 1990, Kapitel 3, S. 56
- [Evo04] Ingrid Gerdes, Frank Klawonn und Rudolf Kruse. *Evolutionäre Algorithmen - Genetische Algorithmen - Strategien und Optimierungsverfahren - Beispielanwendungen*, 1.Auflage, Juli 2004
- [WikiChr] Deutsche Wikipedia, *Chromosom*, <http://de.wikipedia.org/w/index.php?title=Chromosom&oldid=39390217>, Zugriff 27. November 2007
- [WikiEvo] Deutsche Wikipedia, *Evolution*, <http://de.wikipedia.org/w/index.php?title=Evolution&oldid=39333878>, Zugriff 29. November 2007
- [WikiGA] Deutsche Wikipedia, *Genetischer Algorithmus*, http://de.wikipedia.org/w/index.php?title=Genetischer_Algorithmus&oldid=39391300, Zugriff 27. November 2007
- [WikiLog] Deutsche Wikipedia, *Logistik*, <http://de.wikipedia.org/w/index.php?title=Logistik&oldid=39642291>, Zugriff 6. Dezember 2007
- [WikiTSP] Deutsche Wikipedia, *Das Problem des Handlungsreisenden*, http://de.wikipedia.org/w/index.php?title=Problem_des_Handlungsreisenden&oldid=39255834, Zugriff am 25. November 2007
- [WvAnw] Wikiversity, *Kurs: Genetische Algorithmen*, Kapitel 7, http://de.wikiversity.org/w/index.php?title=Kurs:Genetische_Algorithmen/Kapitel_7&oldid=58102, Zugriff 3. Dezember 2007