1 Traveling Salesman Problem und Genetische Algorithmen

Aus der Sicht des Operations Research kann das Konzept der Genetische Algorithmen einfach als eine intelligente zufallsgesteuerte Suche durch einen großen und komplexen Lösungsraum eines Optimierungsproblems verstanden werden. Diese Suche wird in einer gewissen Analogie zur Evolutionsstrategie in der Natur durchgeführt. Der Name *Genetischer Algorithmus* entstammt der Analogie zwischen der Repräsentation einer komplexen Struktur als Vektor von Komponenten und der genetischen Struktur eines Chromosoms. Bei der Züchtung von Pflanzen oder Tieren sollen zum Beispiel Nachkommen erzeugt werden, die verschiedenen gewünschte Eigenschaften haben, Eigenschaften die auf der genetischen Ebene durch die Art und Weise, wie sich die Chromosome der Eltern verbinden, bestimmt werden. Will man hingegegen bessere Lösungen von Optimierungsproblemen erzielen, ist es ein intuitiv naheliegender Ansatz Teilsequenzen von existierenden Lösungen miteinander zu kombinieren. Jede Lösung entspricht dabei einem *Individuum* und die Menge der aktuell vorhandenen Lösungen nennt man *Population.* Genetische Algorithmen wurden von Holland and der Universität von Michigan in den sechziger und siebziger Jahren entwickelt und die erste systematische Abhandlung erschien 1975 in Hollands Buch *Adaption in Natural and Artificial Systems* [Ho75]. Die oben genannte Parallele zur Natur ist nicht exakt, sie genügte Holland aber um diese Methode zur Problemlösung weiterzuverfolgen.

Im Gegensatz zum *Simulated Annealing* und dem *Tabu Search* verwenden genetische Algorithmen verschiedene Lösungen gleichzeitig, aus denen sie durch geeignete Selektion und durch Kombination von Lösungen neue Lösungen erzeugen. Dabei werden in Analogie zur Natur nur die "stärksten" Lösungen überleben (survival of the fittest). Durch einfache genetische Operatoren, wie zum Beispiel *Crossover* und *Mutation*, werden neue Lösungen aus Teilstücken von bereits bestehenden Lösungen erzeugt, so dass sich die bestehende Population stetig verbessert.

1.1 Operation Crossover

Crossover beruht auf einer relativ einfachen Idee. Angenommen wir haben zwei Strings a und b, die aus jeweils 6 Variablem bestehen und die Lösung eines Problems beschreiben. Wir haben also

$$a = (a_1, \dots, a_6) \text{ und } b = (b_1, \dots, b_6).$$

Nun wird zufallsgesteuert ein *Crossover Punkt* aus den Zahlen $1, \dots, 5$ gewählt und eine neue Lösung wird durch Kombination der Teilstücke der Eltern erzeugt. Wenn der Crossove Punkt zum Beispiel 2 ist, so würden die neuen Lösungen lauten:

$$(a_1, a_2, b_3, b_4, b_5, b_6)$$

und

$$(b_1, b_2, a_3, a_4, a_5, a_6)$$

In vielen Anwendungen ist dieser Komponentenvektor (oder Chromosom) einfach ein String betshend aus 0 und 1.

Die Variablen des Strings werden häufig *Gen* genannt, die möglichen Werte der Variablen *Allel* und schließlich die Position der Variablen *Locus*.

Ausserdem wird noch unterschieden zwischen *Genotyp* und *Phänotyp*. Der Genotyp entspricht einer möglichst cleveren Codierung der Lösung eines Optimierungsproblems und der Phänotyp entspricht dem eigentlich daraus dekodierten Ergebnis bzw. dem daraus ableitbaren Funktionswert der Zielfunktion.

Am Beispiel des TSP (n Städte) könnte man z. B. folgendes annehmen: Genotyp ist eine Permutation der Zahlen von $1, \ldots, n$ z. B.

$$[2, 4, 3, 1].$$

Der resultierende Phänotyp ist der Kreis

$$2 \to 4 \to 3 \to 1 \to 2$$

(In diesem Fall ist die Abbildung von Geno- auf Phänotyp also sehr primitiv, es gibt auch andere Fälle, auch wenn das an diesem Beispiel nicht gut zu demonstrieren ist...) Aus dem Phänotyp errechnet sich dann ein bestimmter Zielfunktionswert, der in der Natur der "Fitness" des Individuums entspricht (hier sagt er nur: wie gut bzw. kurz ist der Hamilton-Kreis)

1.2 Operation Mutation

Der zweite häufig verwendete Operator ist die Mutation. Ziel ist, Bereiche des Lösungsraumes zu erreichen, die möglicherweise durch einfache Crossoveroperationen nicht erreicht werden können. Dabei wird jedes Gene eines Strings betrachtet und mit einer kleinen Wahrscheinlichkeit wird sein Allel (d.h. einfach sein Wert) geändert. Zum Beispiel ist die folgende Änderung möglich:

$$(011001) \to (010011)$$

Dabei wurden das dritte und fünfte Allel geändert.

Eine andere Art der Mutation wird für das TSP verwendet. Hier werden einfach wie beim Simulated Annealing zwei Städte vertauscht oder zwei Kanten überkreuzt.

1.3 Operation Selektion

Die Selektion dient dazu, dass nicht allzuviele Lösungen vorhanden entstehen. Das wird dadurch verwirklicht, dass die Lösungen mit schlechtem Zielfunktionswert eliminiert werden.

1.4 Anwendung auf das TSP–Problem

Nachdem die Grundopertoren der genetischen Algorithmen erklärt sind, ergibt sich der genetische Algorithmus für das TSP Problem in kanonischer Weise.

GENETISCHER ALGORITHMUS FÜR TSP:

(1) Erzeuge eine Polpulation von k Individuen. Jedes Individuum entsteht durch eine zufällige Permutation der Zahlen 1 bis n.

(2) Selektion: Berechne die Längen der entsprechenden Rundreisen. Nur ein bestimmter Anteil überlebt (die schlechtesten Individuen sterben ab).

(3) Fortpflanzung/Vererbung:

   - Kopie eines beliebigen Individuums wird erzeugt und Rekombination: zufällige Zusammensetzung eines neuen Kreises aus zwei alten
   - Mutation: Vertauschung zwei Städte oder Überkreuzung zweier Kanten.

(4) Falls Abbruchbedingung nicht erfüllt, gehe zu (2).

Bei der Wahl der zur Rekombination verwendeten Individuen können diejenigen Individuen bevorzugt gewählt werden, die bereits zu einer relativ kurzen Rundreise gehöhren. Darüber hinaus stehen wie beim Simmulated Annealing wieder eine ganze Anzahl von Möglichkeiten zur Verfügung, wie die einzelnen Parameter des Progarmms eingestellt werden können. Hier gilt es wieder, Experimentierfreude und Ideen an den Tag zu legen.

Dabei ist aber ein Punkt zu bedenken. Sowohl beim Simulated Annealing als auch bei den genetischen Algorithmen muß man nichts über das zu lösende Optimierungsproblem wissen, wir verwenden das Verfahren als Black Box und werten die erzeugte Lösung aus. In diesem Punkt liegt aber auch die Schwäche der Verfahren, denn wir nützen die strukturellen Eigenschaften der Probleme natürlich auch nicht aus und so ist eine zielgerichtete Optimierung nicht ohne weiteres möglich.

Literatur

[Ho75] J. H. Holland, *Adaption in Natural and Artifical Systems*, University of Micigan Press, Ann Arbor, 1975.