

Evolutionäre Algorithmen

Sommerakademie Ftan

16. August 2004

Einführung

- *Selektion* führt in der Natur zum Überleben und Fortpflanzen der am “besten” angepassten Lebewesen.
- Idee: Simulation auf dem Computer mit virtuellen Genen zur Lösung von Optimierungsproblemen.
- Entstehung neuer Lösungen algorithmisch durch Kombination und Mutation der bereits vorhandenen, Selektion durch deren Qualität.
- iteratives Verfahren, bis Abbruchbedingung (z. B. Zeitlimit) erreicht.
- Am Ende Ausgabe des besten erzeugten Individuums als Näherungslösung.

Übersicht

1. **Optimierungsprobleme**
2. Natürliche Evolution
3. Evolutionäre Algorithmen
4. Beispiel TSP
5. Bewertung

Optimierungsprobleme

Ein Optimierungsproblem ist gegeben durch:

- eine Menge Ω , der *Suchraum*.
- eine Abbildung $f : \Omega \rightarrow \mathbb{R}$, die *Zielfunktion*.

deren Maximum gesucht ist:

$$\max_{x \in \Omega} f(x) = ?$$

In den meisten Fällen ist auch ein $x \in \Omega$ gesucht, für das der maximale Wert angenommen wird.

Minimierungsprobleme

Oft ist im Gegensatz dazu der minimale Wert der Zielfunktion gesucht:

$$\min_{x \in \Omega} f(x) = ?$$

Dies ist aber äquivalent zu

$$\max_{x \in \Omega} (-f(x)) = ?$$

so dass man ohne Einschränkung Suche nach dem Maximum annehmen kann.

Verallgemeinerungen

Evolutionäre Algorithmen können auch auf ähnliche Probleme angewendet werden, die sich nicht in dieser Form ausdrücken lassen, zum Beispiel

- Verrauschtheit der Zielfunktion
- dynamische Änderung der Zielfunktion
- Anwesenheit von mehreren Zielfunktionen

spezielle Probleme

Für bestimmte Klassen von Optimierungsproblemen existieren spezialisierte Algorithmen. Diese erfordern eine bestimmte Form von Ω und f , zum Beispiel

- *Lineares Programmieren*: der Suchraum hat die Form $\Omega = \{x \in \mathbb{R}^n \mid Ax \leq b\}$ mit einer linearen Zielfunktion $f : \mathbb{R}^n \rightarrow \mathbb{R}, x \mapsto c^T x$. Dies ist eines der bestuntersuchtesten Optimierungsprobleme und mit geeigneten Methoden leicht zu lösen.
- *kombinatorische Optimierung*: $\Omega \subset \{0, 1\}^n$ ist durch eine kombinatorische Bedingung festgelegt, $f : \mathbb{R}^n \rightarrow \mathbb{R}, x \mapsto c^T x$ ist ebenfalls linear. Diese Klasse enthält die schwierigsten Probleme überhaupt, zum Beispiel das *Traveling Salesman Problem*.
- *reeller Suchraum*: $\Omega = \mathbb{R}^n$ und $f : \mathbb{R}^n \rightarrow \mathbb{R}$ jetzt nichtlinear. Wenn die explizite Form von f nicht bekannt ist, sondern nur einzelne Funktionswerte berechnet werden können, müssen numerische Verfahren verwendet werden.

Insgesamt hat man zu unterscheiden zwischen solchen Algorithmen, die auf spezielle Problemstellungen spezifiziert sind, und solchen, die sich auf sehr viele Klassen anwenden lassen. Normalerweise werden spezialisierte Verfahren wesentlich effizienter sein als die allgemeinen, dennoch liefern bei den allgemeinen Ansätzen gerade die evolutionären Algorithmen oft brauchbare Ergebnisse.

Traveling Salesman Problem

Das Standardproblem der kombinatorischen Optimierung ist das *Traveling Salesman Problem*, kurz *TSP*.

Man stelle sich einen Reisenden vor, der aus einer gegebenen Menge von Städten (*Knoten*) jede genau einmal besuchen und zum Ausgangspunkt zurückkehren will. In Abhängigkeit von der Reihenfolge der Städte wird er dafür mehr oder weniger weit unterwegs sein; die Aufgabe lautet jetzt, die kürzeste Rundreise (*Tour*) zu finden.

Statt als Entfernung zwischen zwei Städten kann die *Kantenlänge* als auch Zeit oder Kosten interpretiert werden.

Heuristiken

- In Anwendungen: optimale Lösung nicht gefragt, nur möglichst gute.
- somit Unterscheidung zwischen exakten Algorithmen und *Heuristiken*.
- Evolutionäre Algorithmen als *Metaheuristik*: auf jedes Problem zuschneidbar, liefert aber nur Näherungslösungen.

Übersicht

1. Optimierungsprobleme
2. **Natürliche Evolution**
3. Evolutionäre Algorithmen
4. Beispiel TSP
5. Bewertung

Das Genom

- Jeder Organismus besitzt in jeder Zelle eine Kopie seiner genetischen Information, die in Form des DNA-Moleküls vorliegt.
- Ein Gen ist (mit gewissen Einschränkungen) ein Abschnitt der DNA, der ein Protein kodiert.
- Grosse Teile der DNA sind *nichtkodierend*.
- Ein Gen kann in verschiedenen Ausprägungen, den sogenannten *Allelen*, vorkommen, z. B. ein Allel für schwarze Haare bei einem Gen, das die Haarfarbe bestimmt.

Genotyp und Phänotyp

- *Genotyp*: Gesamtheit der Allele eines Organismus
- *Phänotyp*: äussere Erscheinung des Individuums.
- Genotyp und Umwelteinflüsse bestimmen den Phänotyp.

Selektion

Die Fitness eines Individuums kann über die Anzahl weitervererbeter Allele gemessen werden. Folgen der Selektion:

- Bei konstanten Umweltbedingungen wird sich jede Art eine ökologische Nische suchen und einem optimalen Genpool entwickeln.
- in der Natur: Durch *Koevolution* (andere Arten verändern sich ebenfalls), ist dies nicht der Fall. Somit entwickelt sich die Evolution nicht zu einem Optimum hin, denn die Fitness eines Individuums hängt sehr stark von der Umwelt ab.
- bei Optimierungsproblem: konstante Umweltbedingungen, somit existiert ein Optimum, das gesucht ist.

Rekombination und Mutation

Dies sind zwei Mechanismen, die die genetische Vielfalt erhöhen.

Rekombination: Durchmischung der Genome bei der sexuellen Fortpflanzung. Geschieht in der Natur durch Auswahl eines Chromosoms von jedem Elternteil. Mit geringer Wahrscheinlichkeit können diese durch *Crossing-Over* auch Gene austauschen.

In vereinfachter Form wird dieses Verfahren auch bei den evolutionären Algorithmen angewendet. Durch Anwendung eines *Rekombinationsoperators* auf zwei Individuen wird ein Kindindividuum berechnet. Wie dieser Operator im Detail aussieht, ist stark vom konkreten Optimierungsproblem abhängig.

Mutation: Das Kopieren der DNA geschieht mit einer Fehlerrate von 10^{-10} bis 10^{-11} pro Nukleotid. Viele Mutationen in der Natur finden in nichtkodierenden Bereichen statt, so dass sie keinen Einfluss auf den Phänotyp haben; gibt es jedoch Auswirkungen, so sind diese in den allermeisten Fällen schädlich. Dennoch sind Mutationen der entscheidende Mechanismus zur Entstehung von neuem Genmaterial.

Auch bei den evolutionären Algorithmen dient dieser Mechanismus zur Erzeugung neuer Eigenschaften der Individuen. Dabei kann ein *Mutationsoperator*, der direkt nach dem Rekombinationsoperator angewendet wird, das Genom geringfügig verändern.

Gendrift und Genfluss

Gendrift ist ein Effekt, der vor allem bei kleinen Populationen auftritt. Gemeint ist damit, dass die Häufigkeit eines Allels natürlichen, zufälligen Schwankungen unterliegt; Ist die Population sehr klein, kann das eine oder andere sogar aussterben. Offensichtlich wird die genetische Vielfalt durch diesen Effekt verringert. Dasselbe tritt natürlich auch in der simulierten Evolution auf, so dass man die Population also nicht allzu klein wählen sollte.

Bei verschiedenen Populationen derselben Art, die räumlich voneinander getrennt sind, werden diese sich natürlicherweise etwas unterschiedlich entwickeln. Findet zwischen den Populationen Migration statt, werden sich dadurch die Genpools vermischen; dies bezeichnet man als *Genfluss*. Dieser Effekt erhöht die genetische Vielfalt. Relevant wird er für genetische Algorithmen bei deren Parallelisierung.

Übersicht

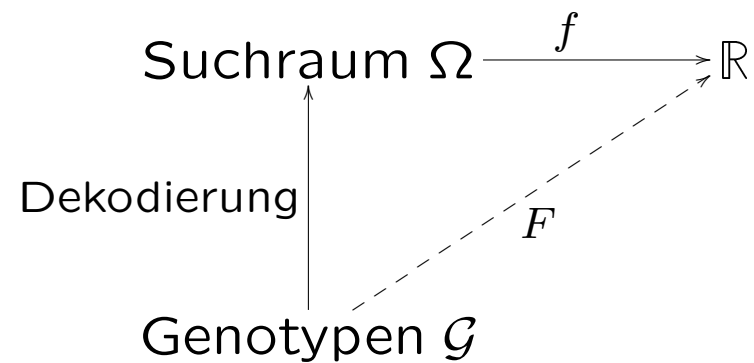
1. Optimierungsprobleme
2. Natürliche Evolution
3. **Evolutionäre Algorithmen**
4. Beispiel TSP
5. Bewertung

Grundprinzip

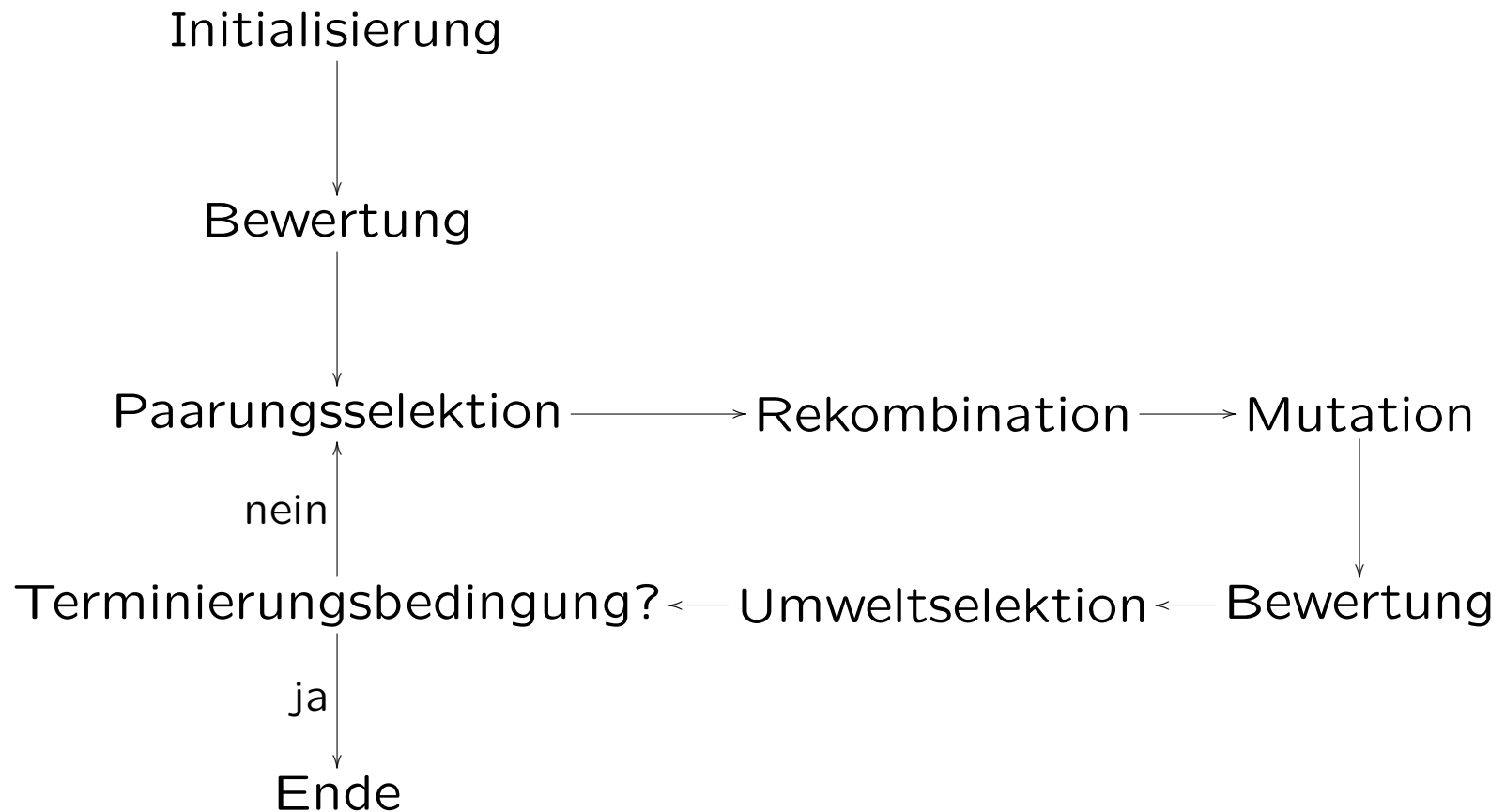
Es stellt sich die Frage, inwieweit man den natürlichen Evolutionsprozess im Detail simulieren sollte. Im folgenden wird der allgemeine Aufbau eines evolutionären Algorithmus vorgestellt, wobei die einzelnen Teile noch nicht näher spezifiziert werden.

- Phänotyp eines Individuums A ist das entsprechende Element aus Ω . Dies bestimmt die Güte des Kandidaten und somit den Selektionsdruck.
- Abgespeichert wird A als Genotyp, woraus der Phänotyp mittels einer *Dekodierungsfunktion* bestimmt werden kann.
- Auf die Genotypen der Population wirken Rekombinations- und Selektionsoperator.

Durch die Dekodierungsfunktion erhält man eine induzierte Bewertung des Genotyps, so dass der Phänotyp gar nicht explizit bestimmt werden muss:



Verlaufsdigramm



Im folgenden werden die einzelnen Schritte kurz erläutert.

Initialisierung Erstelle eine Anfangspopulation von Individuen. Zum Beispiel kann dies durch rein zufällige Genotypen geschehen, man könnte aber auch besser auf das jeweilige Problem zugeschnittene Verfahren verwenden.

Bewertung Berechnung der Zielfunktion für jedes Individuum, so dass anhand dieser Daten eine Selektion durchgeführt werden kann.

Paarungsselektion Wähle die Paare von Individuen aus, die jeweils ein Individuum der nächsten Generation erzeugen sollen. Es existieren verschiedene Auswahlverfahren, die auf unterschiedliche Weisen die Bewertung berücksichtigen; oft werden probabilistische Verfahren verwendet.

Rekombination Wende den Rekombinationsoperator auf jedes Elternpaar an, um ein Kindindividuum zu erzeugen. Oft wird an einer zufälligen Stelle ein Crossing-over durchgeführt. Essentiell für die Leistung des Algorithmus ist, dass der Rekombinationsoperator im Suchraum lokal arbeitet, das heisst dass ähnliche Eltern ein Kind erzeugen, das ihnen ebenfalls ähnlich ist. Ausserdem sollte bei einer guten Bewertung der Eltern auch ein gutes Kind entstehen.

Mutation Jedes Kindindividuum wird jetzt durch den Mutationsoperator möglicherweise noch verändert. Dieser kann zum Beispiel darin bestehen, dass jedes Bit des Genotyps mit einer gewissen (genügend kleinen) Wahrscheinlichkeit invertiert wird. Genauso wie bei der Rekombination ist darauf zu achten, dass die Mutation im Suchraum lokal verläuft, dass sich also der Phänotyp nicht zu sehr verändert.

Umweltselektion Beim Hinzufügen der Kindindividuen zur Population können diese entweder die alte Generation vollkommen ersetzen oder in die Elternpopulation eingefügt werden. Zumindest im zweiten Fall benötigt man einen weiteren Selektionsschritt, da sonst die Populationsgrösse beliebig wachsen würde. Diesemal werden, möglicherweise mit einer probabilistischen Komponente, die schlechteren Individuen ausselektiert.

Übersicht

1. Optimierungsprobleme
2. Natürliche Evolution
3. Evolutionäre Algorithmen
4. **Beispiel TSP**
5. Bewertung

Im folgenden wird der genetische Algorithmus von *Sengoku* und *Yoshihara* vorgestellt.

- Der Genotyp eines Individuums ist eine Permutation aus S_n , in der alle Knoten der Reihenfolge in der Tour nach aufgeführt sind; über die Dekodierung gibt es also jeweils genau $2n$ Genotypen, die zum selben Phänotyp führen.
- Die Elternpaare in der Paarungsselektion werden zufällig ausgewählt.
- Die Elternpopulation wird auf einen vorgegebenen Prozentsatz verringert, wobei der bessere Teil der Population beibehalten wird. Der Rest wird aufgefüllt mit aus Rekombination und Mutation neu erzeugten Individuen, so dass die Populationsgröße konstant bleibt.

- Rekombinationsoperator: Seien die beiden Genotypen $A = (a_1, \dots, a_n)$ und $B = (b_1, \dots, b_n)$ gegeben.
Wähle einen zufälligen Knoten t ; ohne Einschränkung kann $a_1 = b_1 = t$ angenommen werden.
Erstelle das Kindindividuum $C = (c_1, \dots, c_n)$ auf folgende Art: $c_1 = t$, und füge jetzt alternierend Knoten von A und B hinzu, und zwar die Knoten von A nach rechts, die von B ausgehend vom Ende der Permutation nach links. Wenn beides nicht mehr möglich ist, weil ein Knoten schon irgendwo vorkommt, fülle die leeren Stellen mit den übrigbleibenden Knoten in beliebiger Reihenfolge.

- Mutationsoperator: Dieser wird mit einer gegebenen Wahrscheinlichkeit angewendet und besteht aus der Anwendung der *2opt-Heuristik*.

Ein einzelner *2opt*-Schritt besteht aus der Inversion eines Teilstücks der Permutation. Phänotypisch bedeutet das, dass der entsprechende Teils der Tour in umgekehrter Reihenfolge durchlaufen wird.

Es werden so viele *2opt*-Schritte durchgeführt, bis dadurch keine weitere Verbesserung erreicht werden kann; dies wird dadurch überprüft, dass alle Kantenpaare nacheinander getestet werden.

Dieser Mutationsoperator ist sehr stark an das TSP angepasst und ist kein solcher im klassischen Sinne, da er die genetische Vielfalt nicht erhöht. Aufgrund dieser Anpassung liefert der Algorithmus hervorragende Ergebnisse.

Übersicht

1. Optimierungsprobleme
2. Natürliche Evolution
3. Evolutionäre Algorithmen
4. Beispiel TSP
5. **Bewertung**

Zusammenfassend lässt sich sagen, dass evolutionäre Algorithmen ein sehr robustes und vielseitiges Werkzeug zur näherungsweise Lösung von Optimierungsproblemen darstellen.

Bei genügender Anpassung an das spezielle Problem können sehr gute Ergebnisse erhalten werden, mit naiven Ansätzen erhält man noch mittelmässige Ergebnisse. Aber gerade bei Aufgaben, die noch schlecht untersucht oder auf andere Art und Weise nicht zugänglich sind, bieten sich evolutionäre Algorithmen an.

Angewendet werden sie zum Beispiel zur Planung von Stunden- und Arbeitsplänen, aber auch in anderen Feldern der Betriebswirtschaft.