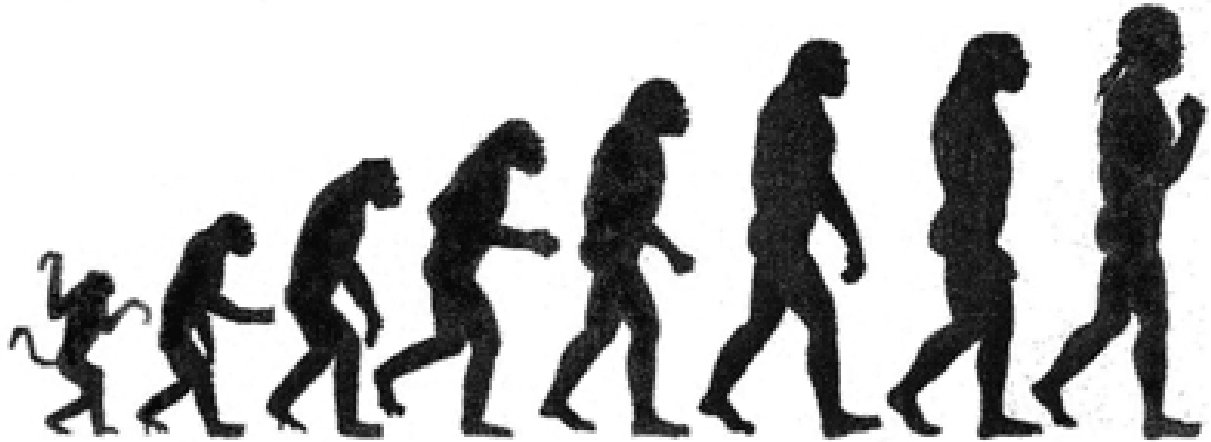


Genetische Algorithmen

Quelle: <http://www.biologie-schule.de/synthetische-evolutionstheorie.php>



1

Inhaltsverzeichnis

Genetische Algorithmen.....	1
Allgemein:.....	2
Anwendungsgebiete.....	2
Motivation	2
Elemente aus genetischen Algorithmus.....	2
Beispiel N Damen-Problem	3
Allgemeines	4
Individuum.....	4
Fitness.....	4
Mapping.....	4
Ablauf	4
Konvergenz von Genetischen Algorithmen.....	6
Anwendungsgebiete.....	7
Wirtschaft.....	7
Forschung	7
Kunst und Musik.....	7
Probleme von Genetischen Algorithmen	7
Stärken und Schwächen	7
Fazit	7

¹ <http://www.biologie-schule.de/synthetische-evolutionstheorie.php>

Allgemein:

Genetische Algorithmen sind Algorithmen, die mit einem Soft-Computing Ansatz zu lösen sind. Soft-Computing werden durch **modellfreie Ansätze**, **Annäherungen** statt exakten Lösungen (die nicht immer ausreichen), **Schnelles** finden einer **brauchbaren** aber nicht exakten Lösung.

Anwendungsgebiete

Lösungsansätze

Analytische Lösung: sehr effizient, aber nur in seltenen Fällen anwendbar

vollständige Durchforstung: sehr ineffizient (=nur kleine Suchbäume)

blinde Zufallssuche: immer anwendbar, (meist nicht effizient)

gesteuerte Suche: Es ist nötig, dass die Funktionswerte, den Elemente des Suchbaumes ähnlich sind.

Parameteroptimierung, bezeichnet das finden eines Parameters, sodass der gegebene reallwertige Funktion ein Optimum annimmt.

Packproblem: Rucksackproblem (Wert des Rucksackes Maximal, bei begrenzten Gewicht)

Wegeproblem: Traveler Salesman Problem (Reinfolge der Ziele), Anwendung Bohren von Platinen

Anordnungsproblem: Positionsfindung von Verteilerknoten (Steinerbäume usw..)

Planungsprobleme: Arbeitspläne, Zeitpläne, Optimierung von Compiler

Strategische Probleme: Gefangenendilemma

Biologische Modellierung: Spinnennetz Bildung

Motivation

Genetische Algorithmen basieren auf der biologischen Evolutionstheorie. Das Prinzip, dass es in jeder Generation oder Iteration eine zufällige Variation gibt, und dass durch die Natürliche Auslese bestimmt wird ob die Veränderung gut war oder nicht.

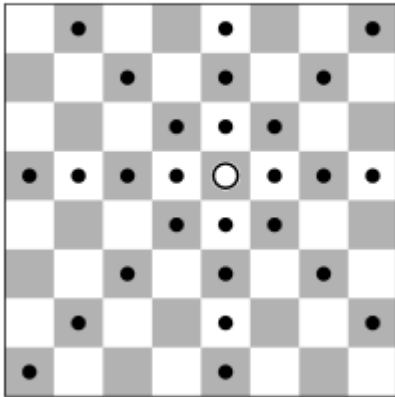
Dieses Prinzip stützt sich eben auf die Diversität (vielfalt), Variation (Mutation), Vererbung (nächste Generation weitergeben), Artbildung (so weit variiert, dass neue Art entsteht), Überproduktion (falls jemand stirbt), Anpassung (wer sich nicht anpasst wird Ausgelesen), blinde Variation (Zufälligkeit), Gradualismus (nur kleine Änderungen pro Iteration / Generation), Vererbung mit Modifikationen (was gut funktioniert behalten wir), ökologische Nischen (Finden Platz zu leben), wachsende Komplexität (pro Generation / Iteration)

Elemente aus genetischen Algorithmus

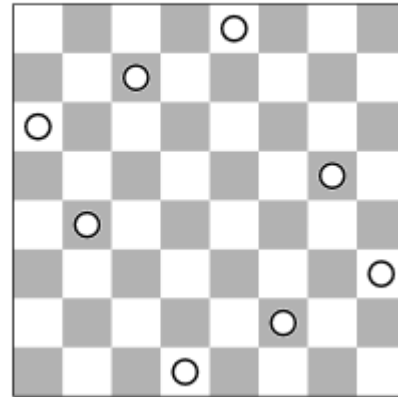
Ein Algorithmus besteht aus einer **Kodierungsvorschrift** (wie werden Lösungskandidaten kodiert), einer Methode, die einen **Anfangspopulation** erzeugt, einer **Bewertungsfunktion** (Bewertung der zu optimierenden Funktion), einer **Auswahlmethode** (Was bleibt was geht), **genetische Operatoren** (Mutationen (Zufall), Crossover (Verbindungen)), **Parameter** (Populationsgröße ...), **Abbruchkriterium**

Beispiel N Damen-Problem

Zugmöglichkeiten einer Dame



Lösung des 8-Damen-Problems



2

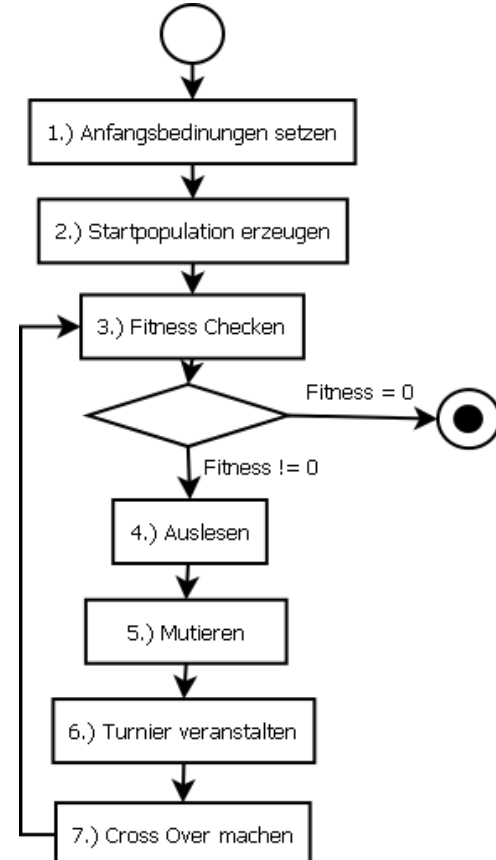
Lösungsansatz Backtracking:

- Die Damen werden zeilenweise, von unten nach oben, platziert.
- Jede Zeile wird
 - Jede Dame von links nach rechts auf die Felder der Zeile gesetzt
 - Prüfung auf Kollision (in tieferen Zeilen)
 - Wenn Ja, dann ein Feld weiter Rechts Prüfen.
 - Wenn Nein nächste Zeile Rekursiv aufrufen
- Wird eine Lösung gefunden, in der keine Kollision stattfindet wird diese ausgegeben.

Lösungsansatz Genetischer Algorithmus³

Der Lösungsansatz dieses Problems geht wie oben beschrieben von einem Abbild von natürlichen Evolution Prozessen aus. Hierbei werden die einzelnen Schachkonstellationen zu Individuen, die eine Evolution Zyklus solange durchgehen und pro Generation immer besser werden, bis man ein Individuum hat, das die Kriterien einer Lösung entspricht. (Dass keine Dame eine andere schlagen kann)

Da eine Generation zur nächsten immer besser wird liegt am Cross Over den die Individuen miteinander vollziehen und dass nur die besten Individuen ausgewählt werden diesen zu machen. Es werden einfach Teile der Eltern an die Kinder weitergegeben, und man hofft dass die besser sind. Wenn das diese Art der Auslese und des Reproduzieren immer weiter gemacht wird hat man am Ende ein Individuum das die Kriterien einer Lösung erfüllt.



² <http://www.borgelt.net/slides/ga.pdf>

³ <http://www.codeproject.com/Articles/12004/Queens-Solution-with-Genetic-Algorithm>

Allgemeines

Individuum

Ein Individuum ist eine mögliche Lösung des Problems, Diese kann Richtig oder auch Falsch sein. In unseren Fall ist ein Individuum eine Konstellation von n Damen auf einen $n \times n$ großen Schachfeld.

Fitness:

Die Fitness eines Individuums ist die Betrachtung wie gut eben jenes Individuum ist und folglich wie gut die Lösung des Problems ist. In unseren Fall gibt die Fitness eines Individuums die maximale Anzahl der Damen an die eine Dame angreifen. Falls in einen Individuum 2 Damen eine Angreifen und auch noch 3 weitere Damen eine weitere Dame Angreifen so ist die Fitness 3.

Damit man aus der Fitness ein Maximierungsproblem macht multipliziert man die so entstandenen Finesse noch mit (-1).

Somit möchte man versuchen die Maximale Fitness von 0 zu erreichen (keine Dame schlägt eine andere = Problem gelöst) und man möchte immer die vorhande Fitness maximieren.

Mapping

Jedes Individuum im System wird als Chromosom dargestellt. Der Genotyp von jedem Chromosom ist ein Vektor der so viele Dimensionen hat wie die Größe des Problems n .

Das Mapping vom Schachfeld zu dem Genotyp passiert indem man Zeilenweise vorgeht. Man schreibt pro Zeile die Spaltennummer, in der die Dame Steht in die Zeile des Vektors mit derselben Zeilennummer, wie die des Schachfeldes, bei dem man gerade die Dame betrachtet hat. Das muss man nur noch für jede Zeile des Schachfeldes machen und man bekommt sein Chromosom.

Hierbei ist es nun nicht möglich, dass 2 Damen in derselben Zeile stehen. Dieser Fakt schränkt auch nicht unseren Lösungsraum ein da es beim n Damenproblem keine Lösung gibt, wo 2 Damen in derselben Zeile stehen.

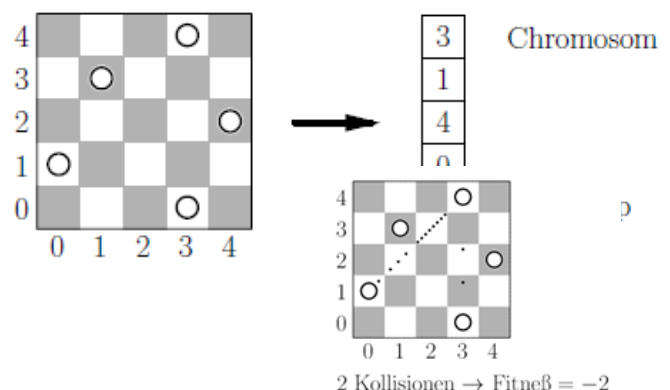
Ablauf

1. Wähle die passenden Parameter:

- Population $[p]$
 $p \in \mathbb{N}$, $p > 0$, $p \bmod 4 = 0$, p = Anzahl der Individuen im System
- Maximale Iterationsanzahl
- Mutation Faktor [5%-10%]
- N Damen $[n]$
 $n \geq 5$, n = Größe des Problems, n Damenproblem

2. Erzeuge die Startpopulation:

- Erzeuge Pro Individuum ein Chromosom zufällig, wobei bei einen Individuum eine Dame in jede Zeile Zufällig gesetzt wird, und nachher in ein Chromosom umgewandelt wird. (Beim Ausprogrammieren wird man natürlich nur pro Vektorzeile eine Zufallszahl von $0-(n-1)$ reinschreiben)



3. Decodiere die Chromosomen und finde die Fitness

- Überprüfe Spalten Abhängigkeiten
- Überprüfe Diagonalen
- Berechne Fitness (max = 0 -> Lösung)

4. Auslese

Falls in die Bevölkerung zu groß ist (Iteration 1-Ende)

- Alle durchgegangen und die Individuen mit der Schlechtesten Fitness rauswerfen, solange bis wir die nenn Bevölkerung erreicht haben.

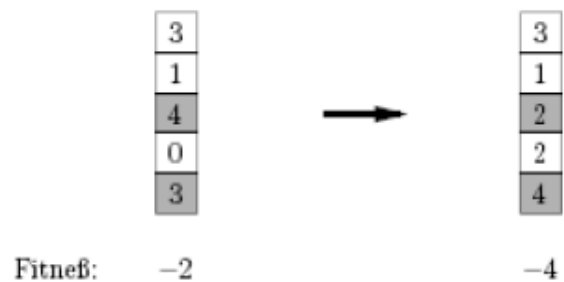
5. Veranstatle Turnier

Das Turnier können $p/2$ Individuen gewinnen (also die Hälfte der Bevölkerung)

- Es wird eine Zufallsentscheidung getroffen wer das Turnier gewinn gewichtet nach der Fitness. Also ist es auch möglich mit einer miesen Fitness zu gewinnen aber die Wahrscheinlichkeit das zu tun ist geringer als mit einer hohen.
- Jetzt haben wir 50% Ausgewählte Individuen und 50 % die nicht Ausgewählt wurden.

6. Mutation

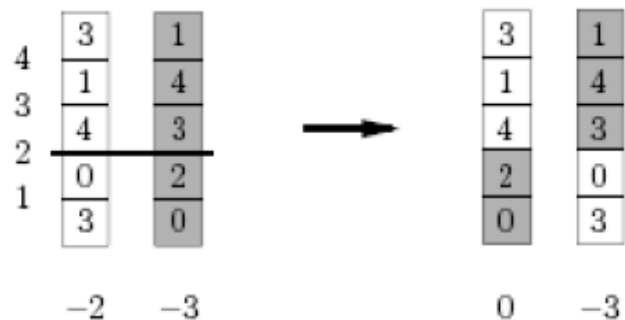
Damit es nicht zum Fall kommt das immer dieselben Chromosomen ausgetauscht werden gibt es die Mutation, die so viele betrifft wie Oben der Parameter genannt wird. Dabei werden einfach x% der Individuen Zufällig viele Zufällige Chromosomen gesetzt.



7. Cross Over

Jetzt nehmen wir nur die Individuen die das Turnier gewonnen haben und lassen sie jeweils Paarweise den Crossover machen.

- Es werden nun 2 Individuen genommen und bei beiden eine Zufällige Linie im Chromosom gezogen. Dies darf nicht am Rand passieren.
- Danach werden 2 neue Individuen erstellt wobei eines den Chromosom Teil der Mutter unterhalb der Linie erbt und den Chromosom Teil über halb vom Vater. Das zweite Individuum erbt vom Vater alles was Unterhalb der Linie ist und von der Mutter den übrigen Teil. (Also einfach die Eltern Kopieren und den Teil über der Linie Austauschen)



Konvergenz von Genetischen Algorithmen

Die Konvergenz von genetischen Algorithmen allgemein festzulegen ist nicht so leicht u.a. weil es auch zu diesen Thema wenig theoretische Resultate gibt. Man kann jedoch in unseren Fall beweisen (wo das beste Individuum in der Population bleibt), dass

Konvergenzsatz: Ist \vec{x}^* das globale Maximum einer Optimierungsaufgabe

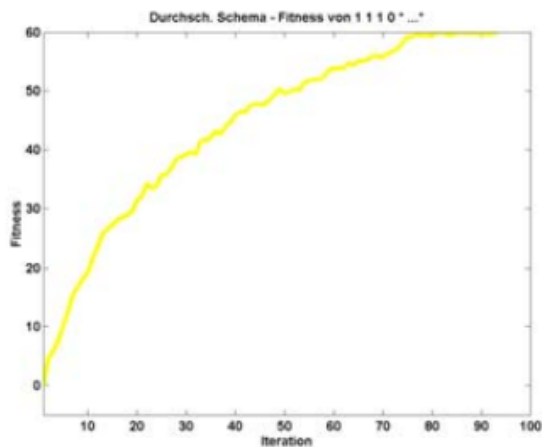
$$\max_{\vec{x}} F(\vec{x})$$

und \vec{x}_n^b das **beste Individuum** in der n -ten Generation eines Genetischen Algorithmus (Genetischer Algorithmus mit „elitärer Strategie“).

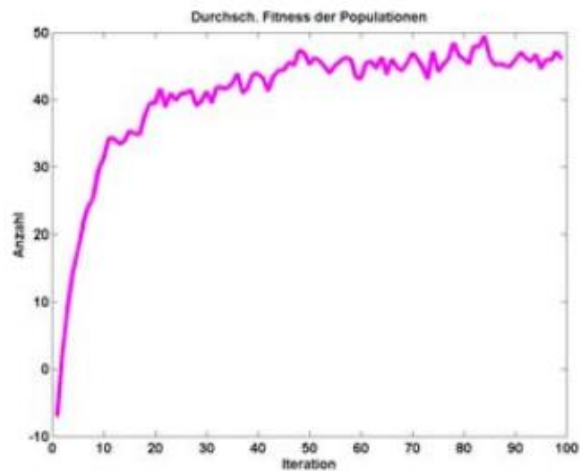
Dann gilt:

$$\lim_{n \rightarrow \infty} P(F(\vec{x}_n^b) = F(\vec{x}^*)) = 1. \quad (140.1)$$

4



*Durchschnittliche Fitness
des „guten“ Schemas*



Durchschnittliche Fitness der Population

Diese Resultate lassen sich so Interpretieren dass im Grenzfall das beste Individuum einer Population mit der Wahrscheinlichkeit 1 die Fitness des globalen Optimums hat(Lösung des Problems ist) = stochastischer Konvergenz.

Also man kann sagen dass bei Genetischen Algorithmen die eine elitäre Strategie aufweisen

- Die mittlere Fitness der Population steigt
- Fitte Konfigurationen breiten sich exponentiell aus.
- Kreuzung und Mutation sorgen dafür, dass auch nicht nur bekannte Fitte Konfigurationen genommen werden und so können noch Fittere Konfigurationen gefunden werden.
- Die Konvergenz ist nicht gesichert
- Tatsächliche Konvergenz kann nur mit Experiment ermittelt werden
- Genetische Algorithmen können lokale Optima wieder verlassen.

⁴ <http://www.iwi.hs-karlsruhe.de/~lino0001/skripte/IntelligenteSysteme/Vortrag%20Stefano%20Cagnoni/AuszugGenAlg.pdf>

Anwendungsgebiete

Wirtschaft

Evolutionäre Algorithmen werden zum Überprüfen ob ein Prototype funktioniert oder zum Optimieren desselben eingesetzt. Beispielsweise wird die Geschwindigkeit von Mirco Prozessoren, Stromverbrauch bei Handys mit Genetischen Algorithmen untersucht. Man kann sie auch für Aktienmarktmodelle einsetzen und bei Portfolios um den Gewinn und minimales Risiko zu optimieren.

Forschung

Evolutionäre Algorithmen werden hier u.a. für Big Data Analysen vor allem in der Molekularbiologie eingesetzt. Damit werden Sequenzanalysen (DNA), Sequenzalignment (Verwandtschaftmodelle), Phylogenetischen Bäumen(Beziehung zwischen verschiedenen Arten z.B. von Lebewesen), Proteinstrukturvorhersage (Dient zur Bestimmung eines Protein) verwendet.

Kunst und Musik

Man kann Evolutionäre Algorithmen zum Entwurf von Tonfolgen einsetzen die den Menschen gefallen. So kann man bei Musik analysieren ob es den Menschen allgemein gefallen könnte oder eine Hilfe beim erschaffen von Musikstücken sein.

Probleme von Genetischen Algorithmen

Bei Genetischen Algorithmen hat man bei größeren Probleme eine Langsame Konvergenz, da der Algorithmus zwar durch Mutation aus Lokalen Maxima herauskommt aber das eben nur sehr langsam. Um den entgegen zu wirken kann man Subpopulationen betreiben(Island Principe) das eben nicht jede Generation die Gene austauscht sondern in Abständen und dann nicht alle sondern nur ein Paar Individuen

Stärken und Schwächen⁵

- Bei Suchen nach einen globalen Optimum funktioniert es gut, da der ganze Lösungsraum durchsucht wird
- Man kann genetische Operatoren hinzufügen um Metawissen an den Genetischen Algorithmus weiter zu geben und dessen Qualität oder Geschwindigkeit zu erhöhen
- Einfache Implementierung für komplexe Probleme
- Nicht effizient bei suche nach lokaler Optima
- Keine Möglichkeit die Dauer der Suche vorherzusagen

Fazit⁶

Genetische Algorithmen sind hervorragend geeignet für komplexe Probleme, die man nicht einfach direkt lösen kann. Man definiert einfach eine Fitness Funktion und überlässt den Algorithmus die Aufgabe. Das einzige was schwierig ist, ist die Definition der Fitness Funktion und manchmal das ausrechnen derselben.

⁵ <http://www.ke.tu-darmstadt.de/lehre/archiv/ss06/se-spiele/slides/GenetischeAlgorithmen-Becher.pdf>

⁶ http://www-e.uni-magdeburg.de/harbach/genetische_algorithmen/genetische_algorithmen.pdf