

# Evolutionäre Optimierungsalgorithmen

FEDERICO RAMÍREZ VILLAGRANA

Universität Hamburg

Methodenkompetenz - Einführung in das wissenschaftliche Arbeiten

Dozent: Dr. Andreas Günther

## Zusammenfassung

*In diesem Paper wird es darüber diskutiert, sowohl was Optimierung ist und warum ist es schwer, als auch was evolutionärer Algorithmen (EA) sind und wie, warum, und wann sind sie hilfreich in dem Optimierung-Bereich. Es wird auch im Detail erklärt, wie EA funktionieren und warum sind sie als teil des künstlichen Intelligenz betrachtet. Es werden auch einige bekannten EA präsentiert und oberflächlich erklärt. Am ende wird es darüber diskutiert, was für Vor- und Nachteile die EA haben.*

## I. EINLEITUNG

Das Problem des Handlungsreisenden (engl. traveling salesman problem oder TSP) ist ein sehr bekanntes und studierte Problem in dem Informatik-Bereich und es geht wie folgendes:

Es muss eine Reihenfolge für den Besuch mehrerer Orte so gewählt sein, dass keine Station außer der ersten mehr als einmal besucht wird, die gesamte Reisestrecke des Handlungsreisenden möglichst kurz, und die erste Station gleich der letzten Station ist. [wikib]

Sei  $n$  die Anzahl der Stationen, es gibt  $(n - 1)!$  mögliche Lösungen für das TSP. Das heißt, dass für  $n = 4$  es gibt 6 mögliche Lösungen. Es ist nicht schwer einen brute-force Ansatz zu verfolgen für ein Problem, das nur 6 Lösungen hat. Aber wenn  $n = 50$  es gibt circa  $6,1 \times 10^{62}$  Lösungen.

Folgendes ist hilfreich, um diese Anzahl in eine Perspektive zu setzen: Das Universum ist circa 15 Milliarde Jahre alt, das ist  $4,7 \times 10^{17}$  Sekunden. Wenn es eine Billion Rechner gäbe, die jeder einzeln seit dem Anfang des Universums eine Billion Lösungen pro Sekunde berechnete, bisher wären nur  $4,7 \times 10^{41}$  Lösungen berech-

net worden.

Das TSP ist nur ein Beispiel von vielfältigen Probleme, die zu den kombinatorischen Problemen gehören, das heißt, Probleme für die einfach keine brute-force Ansatz möglich ist. Es ist in diesem Fall, dass evolutionärer Algorithmen (EA) hilfreich sind.

EA sind eine gute Werkzeug um gute Lösungen zu finden. Natürlich können wir nicht sicher sein, dass wir das beste Lösung gefunden haben, außer wenn wir jeder mögliche Lösungen berechnen haben. Aber wie es mit dem TSP-Beispiel gezeigt ist, es ist nicht immer möglich, alle mögliche Lösungen zu finden.

## II. STAND DER FORSCHUNG

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ip-

sum dolor sit amet, consectetur adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

### III. OPTIMIERUNG

**I**N dem Mathematik-Bereich, Optimierung bedeutet die Findung von Parametern eines Systems, die ein bestmögliches Ergebnis erzielen. [wika] Wir können diese Findung von Parametern auch wie Folgendes definieren: Die Auswahl von die bestmögliche Lösung eines Problems von einer Menge mögliche Lösungen.

In Optimierungsprobleme wird eine Zielfunktion (engl. objective function) definiert, die entweder maximiert oder minimiert werden soll. Die Zielfunktion wird „Kosten-Funktion“ oder „Fitness-Funktion“ in Minimierungs- und Maximierungsprobleme bzw. genannt.

Sei  $f(x)$  ein Zielfunktion,  $x$  ist ein Vektor und wird „Entscheidungsvariable“ genannt. Die Anzahl von Elementen in  $x$  wird die Dimension des Problems genannt. Die Domäne der Zielfunktion repräsentiert die mögliche Lösungen des Problems.

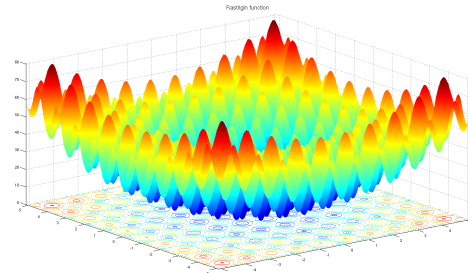
Das Ziel der Optimierung ist es denn, die bestmögliche Lösung des Problems zu finden, das heißt, einen Wert für die Entscheidungsvariable zu finden, sodass die Zielfunktion einen maximal bzw. minimal Wert ergibt.

Es gibt verschiedene Klassifizierungen oder Arten von Optimierung:

- **Eingeschränkt Optimierung.**  
Die Entscheidungsvariable kann nicht irgendwelcher Wert nehmen. Es gibt Einschränkungen dafür.
- **Multi-objective Optimierung.**  
Das Problem besteht aus vielfältigen, voneinander unabhängig Zielfunktionen.

- **Multi-modal Optimierung.**  
Die Zielfunktion(en) hat/haben mehrere Minima bzw. Maxima. Abbildung 1 zeigt eine multi-modale Funktion.

**Abbildung 1:** Rastrigin function. Quelle: [https://commons.wikimedia.org/wiki/File:Rastrigin\\_function.png](https://commons.wikimedia.org/wiki/File:Rastrigin_function.png)



Eine der wichtigste und schwerste Teile der Optimierung ist es, eine geeignete Zielfunktion zu definieren, die alle die wichtige zu optimieren Faktoren berücksichtigt. Probleme im wirklichen Leben sind normalerweise eingeschränkt, multi-Objective und multi-modale, oder mit eine hoch Anzahl von Dimensionen. Wegen diesen Eigenschaften der Zielfunktionen und der Optimierungsprobleme ist es schwer eine gute Lösung mittels traditionelle Vorgänge zu finden. EA sind aber eine gute Möglichkeit, um diese Art von Probleme in angemessene Zeit zu lösen.

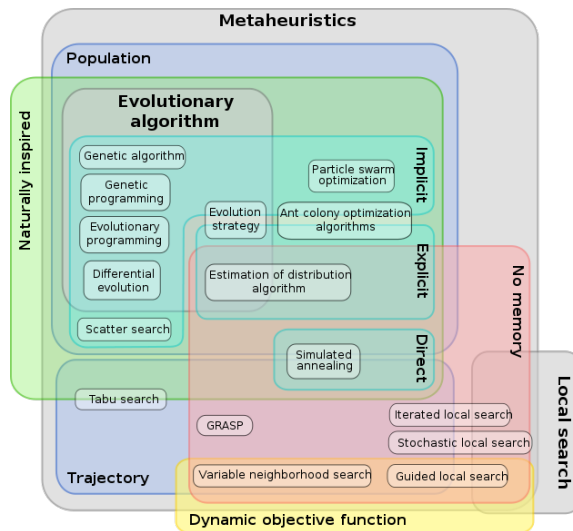
### IV. WAS IST EIN EVOLUTIONÄRER ALGORITHMUS?

**D**er EA-Bereich ist ziemlich neu, deswegen gibt es bisher keine allgemein akzeptierte Definition von evolutionäre Algorithmen.

EA sind zwar als Teil des künstliche Intelligenzes (KI) betrachtet, aber genau wo sie in Verbindung mit anderen KI-Methoden stehen, und was die EA-Bereich beinhaltet, ist von dem Autor abhängig. Abbildung 2 zeigt eine von mehrere mögliche Klassifikationen von KI-Methoden in Bezug auf Metaheuristics (welche

über den Rahmen dieses Papiers hinausgehen).

**Abbildung 2:** Klassifikation von Metaheuristics.  
Quelle: [https://en.wikipedia.org/wiki/File:Metaheuristics\\_classification.svg](https://en.wikipedia.org/wiki/File:Metaheuristics_classification.svg)



In Abbildung 2 können wir es auch sehen, wie der Autor betrachtet Particle Swarm Optimization (PSO) und Ant Colony Optimization (ACO) nicht als EA, trotzdem gibt es andere Autoren (wie z. B. [Sim13]), die diese Algorithmen genau für EA halten.

In diesem Paper wird folgende Definition für EA genommen: Ein Algorithmus, der die Lösung eines Problems durch viele Iterationen entwickelt.

### i. Eigenschaften der EA

Folgende sind die Hauptmerkmale der EA:

- Eine Zielfunktion.
- Eine - so genannte - Bevölkerung von mögliche Lösungen.  
Eine Menge von Darstellungen der Lösungen (Elemente der Domäne der Zielfunktion) des Problems, die mittels der Algorithmus und mehrere Iterationen verarbeitet und hoffentlich verbessert

werden.

- Vorgänge, die die Elementen der Bevölkerung betreffen und verändern.  
Diese können der Algorithmus selber oder ein äußern Verfahren sein.

Die meisten EA sind aus die Natur inspiriert, das heißt, dass Wissenschaftler Ereignisse, Systeme, und Mechanismen von der Natur beobachten und danach versuchen einigermaßen, sie mit Algorithmen zu reproduzieren. Sie werden „evolutionär“ genannt, denn sie sind ursprünglich aus der Evolution inspiriert. [Hol75]

Wie vorher gesagt, EA sind als Teil des KI betrachtet, das ist weil EA emulieren Eigenschaften von intelligente Systeme aus der Natur.

Nach [Sim13] sind die folgende Eigenschaften notwendig, um ein System als intelligent zu betrachten:

- Adaptation.  
Ein intelligentes System muss in hohem Maße an unvorhersehbare Änderungen anpassbar sein. Lernen ist daher unerlässlich.
- Zufälligkeit.  
Obwohl Zufälligkeit meistens als eine schlechte Eigenschaft betrachtet wird, es ist in gewissem Maße notwendig, damit neue Lösungen eines Problems gefunden sein können.
- Kommunikation.  
Eine einzige Ameise ist nicht wirklich als intelligent betrachtet, aber eine Ameisenkolonie ist (dank Kommunikation durch Pheromone) als eine super-intelligent Einheit überlegt.
- Rückmeldung.  
Ein System kann sich nicht anpassen und verbessern, wenn es seine Umgebung nicht erkennen und darauf reagieren kann. Auch um zu lernen, muss ein System seine Fehler (mittels Rückmeldung) erkennen.

- Erkundung.  
Das heißt neue Lösungen finden, neue Wege erkunden.
- Ausbeutung.  
Im Gegenteil zu Erkundung, Ausbeutung heißt, schon bekannten Lösungen und Wege ausnutzen.

In nächster Sektion werden verschiedene Beispiele von EA gezeigt und in jeder Beispiel wird es erwähnt, welcher Teil des Algorithmus welche Eigenschaft des Intelligenzes versucht zu emulieren. Es ist auch wichtig zu sagen, dass nicht alle Eigenschaften des Intelligenzes von jeder EA repliziert sind.

## V. EA BEISPIELE

Heutzutage gibt es aber viele EA und jedes Jahr werden neue entwickelt. In dieser Sektion werden nur ein paar der bekanntesten und wichtigsten erwähnt und oberflächlich erklärt.

### i. Genetic Algorithmen - GA

Genetic Algorithmen (GA) sind ursprünglich in [Hol75] präsentiert und danach in [Gol98] weiter entwickelt. Der Name „Genetic Algorithms“ bezieht sich mehr auf eine Familie von Algorithmen als auf einen einzigen Algorithmus.

GA sind die erste vorgestellte, bekannteste, und am meistens verwendeten EA. Ursprünglich waren sie dafür entwickelt, um adaptierbare Systeme zu studieren. Sie sind Simulationen der natürlichen Selektion. Daher sind sie genau aus der Evolution inspiriert.

Es gibt sechs Hauptteile eines GAs:

#### 1. Codierung der Lösungen.

Als erstes muss man eine geeignete Codierung für die mögliche Lösungen finden. Das heißt, eine Darstellung für die Lösungen finden, die einfach zu verarbeiten ist. Eine die am meistens verwendeten Codierungen ist eine Bit-Folge. In dieser Codierung jedes Bit wird „Gen“ genannt.

#### 2. Ursprüngliche Bevölkerung.

Eine ursprüngliche Bevölkerung muss initialisiert werden. Diese Initialisierung kann entweder zufällig oder auf vorher Kenntnisse basiert sein. Wie viele Elementen die Bevölkerung hat, ist ein wichtige Parameter. Je mehrere Elemente desto eine bessere Wahrscheinlichkeit, um gute Lösungen zu finden.

#### 3. Fitnessfunktion.

Die Zielfunktion des Problems.

Der Wert, den die Zielfunktion ergibt, wenn sie für den Wert ausgewertet wird, den ein Element der Bevölkerung darstellt, wird die „Fitness“ des Elements genannt.

#### 4. Auswahl von Einzelnen.

Zwei Elemente der Bevölkerung werden ausgewählt, um ein neue Element herzustellen. Das Verfahren um die Elemente auszuwählen kann entweder zufällig oder auf die Fitness der Elemente basiert sein. Die ausgewählte Elemente werden „Eltern“ genannt.

#### 5. Crossover.

Dies ist das wichtigste Teil eines GA. Es definiert wie das neue Element aus die Eltern erstellt wird. Das heißt, welche Gene wird das neue Element von jeden Eltern teil erben.

#### 6. Mutation.

Einige zufälligerweise-ausgewählt neue erstellt Elemente werden mutiert sein. Das heißt, dass einige Bits gekippt werden. Welche Elemente werden mutiert und wie viele und welche Bits werden gekippt, kann mittels verschiedene Verfahren entschieden werden.

Bei GA ist Adaptation anwesend, denn diese Art von Algorithmen ist Problem-unabhängig. Mit anderen Wörter, wenn es eine geeignete Darstellung der Lösungen gibt, kann das Problem mit GA optimiert wird.

**Abbildung 3:** Wesentlich genetic Algorithmen pseudo-Code. Quelle: [Sim13]

---

```

Parents ← {randomly generated population}
While not(termination criterion)
  Calculate the fitness of each parent in the population
  Children ← ∅
  While |Children| < |Parents|
    Use fitnesses to probabilistically select a pair of parents for mating
    Mate the parents to create children  $c_1$  and  $c_2$ 
    Children ← Children  $\cup \{c_1, c_2\}$ 
  Loop
  Randomly mutate some of the children
  Parents ← Children
Next generation

```

---

In Teil 2 sowie in Teile 4 bis 6 kann Zufälligkeit verwendet werden, je nachdem welche Verfahren bzw. ausgewählt werden.

In der Mutation-Teil, eine hoch Wahrscheinlichkeit um Elemente zu mutieren heißt viel Erkundung und weniger Ausbeutung und umgekehrt.

## ii. Particle Swarm Optimization - PSO

Particle Swarm Optimization (PSO) Algorithmus wird ursprünglich in [JK95] und in [YS98] präsentiert.

## iii. Differential Evolution - DE

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

## VI. DISKUSSION

**L**orem ipsum dolor sit amet, consectetur adipiscing elit. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

## VII. FAZIT

**L**orem ipsum dolor sit amet, consectetur adipiscing elit. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

## QUELLENVERZEICHNIS

[Gol98] David Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*.

ning. Reading. MA: Addison-Wesley Professional, 1998.

- [Hol75] John Henry Holland. *Adaptation in Natural and Artificial Systems*. MIT Press, 1975.
- [JK95] Russell C. Eberhart James Kennedy. *Particle Swarm Optimization*, volume IV. Proceedings of IEEE International Conference on Neural Networks, 1995. pp. 1942 - 1948.
- [Sim13] Dan Simon. *EVOLUTIONARY OPTIMIZATION ALGORITHMS - Biologically-inspired and Population-based Approaches to Computer Intelligence*. John Wiley and Sons, Inc., 2013.
- [wika] Optimierung. <https://de.wiktionary.org/wiki/Optimierung>. Accessed: 03.01.2019.
- [wikb] Problem des handlungsreisenden. [https://de.wikipedia.org/wiki/Problem\\_des\\_Handlungsreisenden](https://de.wikipedia.org/wiki/Problem_des_Handlungsreisenden). Accessed: 03.01.2019.
- [YS98] Russell C. Eberhart Yuhui Shi. *A modified particle swarm optimizer*. Proceedings of IEEE International Conference on Evolutionary Computation, 1998. pp. 69 - 73.