

Semesterarbeit

# Evaluation von verschiedenen Ansätzen für Evolutionäre Algorithmen

Adrian Schmid

Zürich, 01.01.2014

## Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>2</b>
1.1	Aufgabenstellung . . . . .	2
1.2	Ausgangslage . . . . .	2
1.3	Bla . . . . .	2
<b>2</b>	<b>Planung</b>	<b>2</b>
2.1	Projektplanung . . . . .	2
2.2	Endliche Automaten . . . . .	2
2.3	Evolutionäre Algorithmen . . . . .	2
<b>3</b>	<b>Umsetzung</b>	<b>2</b>
3.1	Automaten . . . . .	2
3.2	EA mit fester, globaler Problemmenge . . . . .	2
3.3	EA mit mutierender, globaler Problemmenge . . . . .	2
3.3.1	Probleme mit mutierenden Problemengem . . . . .	2

# **1 Einleitung**

## **1.1 Aufgabenstellung**

## **1.2 Ausgangslage**

## **1.3 Bla**

# **2 Planung**

## **2.1 Projektplanung**

## **2.2 Endliche Automaten**

## **2.3 Evolutionäre Algorithmen**

# **3 Umsetzung**

## **3.1 Automaten**

## **3.2 EA mit fester, globaler Problemmenge**

## **3.3 EA mit mutierender, globaler Problemmenge**

### **3.3.1 Probleme mit mutierenden Problemmengen**

Der erste Ansatz zur Mutation von Problemmengen war - ähnlich der Mutation von Automaten - wie folgt aufgebaut:

1. Beim durchgehen und testen der einzelnen Automaten wird gezählt welches Problem wie oft korrekt gelöst wurde
2. Die Probleme werden nach dem Zähler sortiert
3. Die schwierigere Hälfte (schlechter gelöst) wird beibehalten
4. Die einfachere Hälfte wird durch neue zufällige Probleme ersetzt

Dies führte dazu, dass die Problemmenge schneller schwieriger wurde, als die Automaten besser. Warum das zu einem Problem wird ist folgend anhand eines Beispiels erläutert:

Wir haben folgendes Setup:

- 200 Automaten als Lösungskandidaten
- 1000 Probleme in der Problemmenge
- Algorithmus mit globaler, mutierender Problemmenge, welche sich wie oben beschrieben verhält.

Der Algorithmus wurde gestartet und es wurden Histogramme der Ergebnisse der Fitnessfunktion der einzelnen Lösungskandidaten erstellt. Die Fitnessfunktion gibt an wieviele der 1000 Probleme der Lösungskandidat richtig gelöst hat.

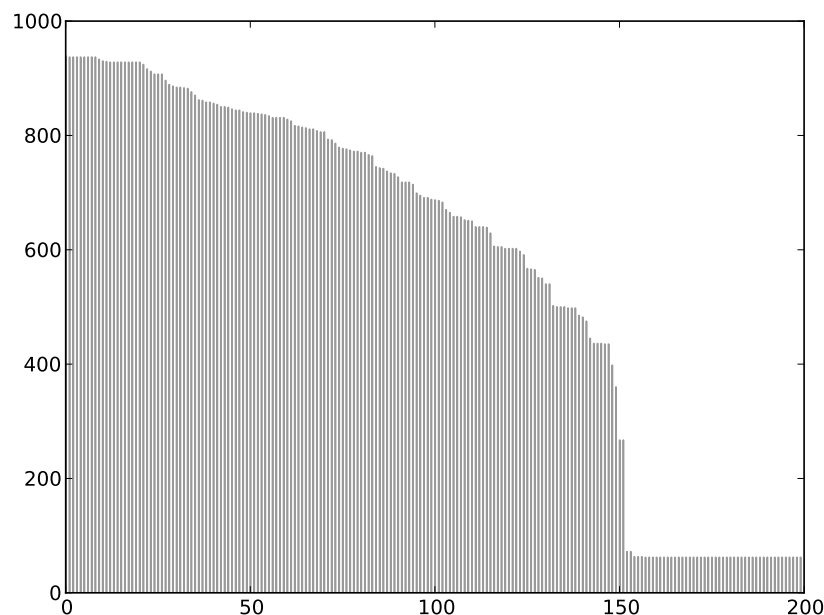


Abbildung 1: Fitnessfunktion der Lösungskandidaten Ausgangslage

Beim ersten Histogramm sieht man den Wert der Fitnessfunktion von 200 zufällig generierte Automaten welche versuchen 1000 zufällig generierte Probleme zu lösen. Einige Automaten schneiden besser ab, einige schlechter. Aber die Verteilung ist zufällig.

Nach einer Iteration lösen unsere Automaten das Ganze schon viel schlechter. Das liegt daran, dass die 500 leichtesten Probleme mit grosser Wahrscheinlichkeit von einem Grossteil der besseren Automaten gelöst werden konnten und eben diese Probleme wurden

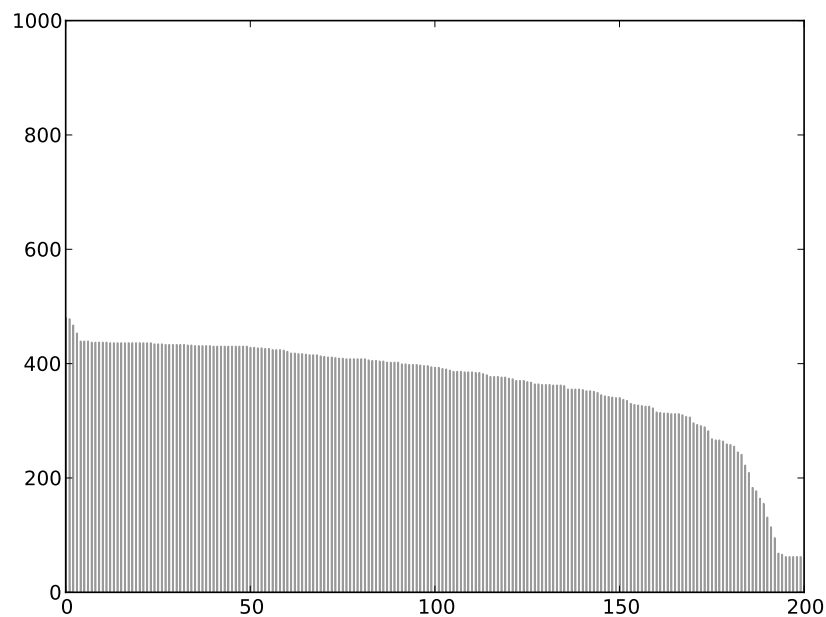


Abbildung 2: Fitnessfunktion der Lösungskandidaten nach einer Iteration

nun entfernt und durch neue, tendenziell schwierigere Probleme ersetzt. Die Automaten hingegen können sich in einer Iteration kaum verbessern.

Nach fünf Iterationen sieht man wozu das Ganze führt: Die Probleme sind nun so schwierig, dass um 95% der Probleme von keinem unserer Automaten mehr gelöst werden kann. Dies führt dazu, dass keine Unterschiede zwischen guten und schlechten Automaten mehr feststellbar sind was wiederum dazu führt, dass der Algorithmus nicht konvergiert.

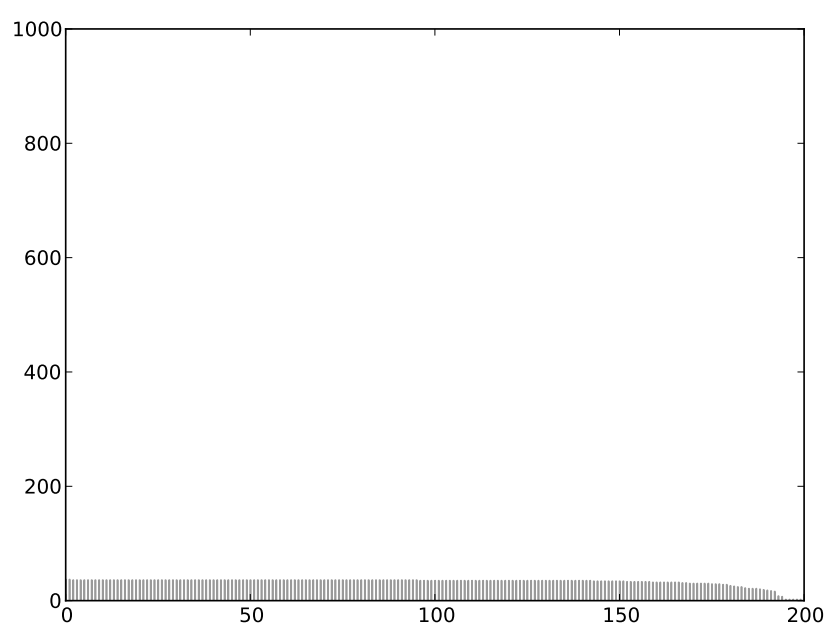


Abbildung 3: Fitnessfunktion der Lösungskandidaten nach fünf Iterationen

## Abbildungsverzeichnis

1	Fitnessfunktion der Lösungskandidaten Ausgangslage . . . . .	3
2	Fitnessfunktion der Lösungskandidaten nach einer Iteration . . . . .	4
3	Fitnessfunktion der Lösungskandidaten nach fünf Iterationen . . . . .	5

## Tabellenverzeichnis