

# Eine Übersicht über Crossover-Operationen für genetische Algorithmen Seminar Organic Computing

Gerald Siegert  
Matrikelnummer: 1450117

Universität Augsburg  
Lehrstuhl für Organic Computing  
[student@organic-computing.org](mailto:student@organic-computing.org)

**Abstract.** Zusammenfassung des Inhalts des Papers in ca. 200 Wörtern.

## 1 Einführung in genetische Algorithmen

Genetische Algorithmen sind genauso wie andere Evolutionäre Algorithmen im Allgemeinen aus der Biologie übernommen worden. Wie der Name schon aussagt, basieren sie auf dem Prinzip der Evolution, bei der basierend auf einer Ausgangspopulation möglicher Lösungen neue Kinder erzeugt werden, welche dann die Vorfahren in der Population verdrängen. Welche Vorfahren, oder gar die erzeugten Kinder, dabei konkret verdrängt werden, entscheidet sich basierend auf einer Fitness-Funktion, bei der die gefundenen Lösungen der Population Evaluert werden und anschließend die nur die besten in der Population verweilen dürfen.



Fig. 1: Grundlegender Ablauf eines genetischen Algorithmus

Die wichtigen Parameter eines GA selbst sind zum einen die Selektion der Gene, deren Crossover zur Erzeugung neuer Kinder, sowie die Durchführung anschließender Mutationen. In dieser Seminararbeit soll ein kleiner Überblick über verschiedene Crossover-Operationen gegeben werden, welche maßgeblich über die Qualität und Effizienz eines GA entscheiden.

## 2 Klassifizierungen von Crossover-Operationen

Klassifizierung von Crossover

## 3 Eindimensionale Repräsentation

Eindimensionales

### **3.1 Binäre Codierung**

Binär

### **3.2 Codierung als Ganzzahlen**

Integer

**Operationen für ganzzahlige Werte** Integer, die nicht als Binär gehandhabt werden

**Operationen für Permutationen** Permutationen von Integer-Werten (zB TSP)

### **3.3 Codierung als Fließkommazahl**

Fließkommazahlen

### **3.4 Codierung als Zeichenkette**

String-Codierungen

## **4 Mehrdimensionale Repräsentation**

Mehrdimensionale

### **4.1 Codierung als Baum**

Bäume und deren nutzen

### **4.2 Codierung als Array**

Array und deren Nutzen

### **4.3 Weitere Codierungen für mehrdimensionale Daten**

Kurz weiteres wie Matrizen und modularisierte Codierung

## **5 Anwendungsspezifische Codierung der Daten**

Kurz anwendungsspezifisches

## **6 Universale Crossover-Operationen**

Kurz auf weitere, universal einsetzbare Operationen eingehen (besser am Anfang?)

## **7 Zusammenfassung und Ausblick**

Kurze Zusammenfassung

## **References**

1. Müller-Schloer, C., Schmeck, H., Ungerer, T. (eds.): Organic Computing - A Paradigm Shift for Complex Systems. Birkhäuser (2011)

## 8 Motivation

Einführung ins Thema. Was bestehen für Probleme, wie soll das gelöst werden?

Wieso braucht man vorgestellte Technik/System/Algorithmus?

## 9 Stand der Technik

Wie andere Verfahren das Problem zu lösen versuchen. [1]

## 10 Hauptteil

### 10.1 Grundlagen

Text zu Fig. 2. Siehe Formel 1

**Advantages and Challenges** Eine Subsubsection.



Fig. 2: Ein Beispielbild.

$$\bar{e} = \frac{1}{N} \sum_{i=1}^N |f_i - x_i| \quad (1)$$

## 11 Evaluation



(a) Beispielbild 1



(b) Beispielbild 2

Fig. 3: Zwei Beispielbilder.