

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

Abstract. Crossover-Operationen sind ein wesentlicher Teil von Genetischen Algorithmen und sind maßgeblich für deren Effizienz. Daher soll ein kleiner Überblick über verschiedene Crossover-Operationen und deren Klassifizierung gegeben werden. Zunächst werden eindimensionale Repräsentationen betrachtet. Dabei werden Repräsentationen als Binärwerte, Ganzzahlen bzw. entsprechende Permutationen, Fließkommazahlen und Strings erläutert, wann welche Repräsentation geeignet ist und dafür optimierte Crossover-Operationen aufgezeigt. Ebenso betrachtet werden mehrdimensionale Repräsentationen wie Bäume und Arrays. Es wird zudem eine kleine Übersicht über weitere mehrdimensionale Repräsentationen gegeben. Ebenso wird auch darauf eingegangen, wann es geeignet ist, anwendungsspezifische Codierungen zu nutzen und anzuwenden. Ebenso werden zudem einige universell nutzbare Crossover-Operationen aufgezeigt, die nicht an eine spezielle Repräsentation gebunden sind.

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.

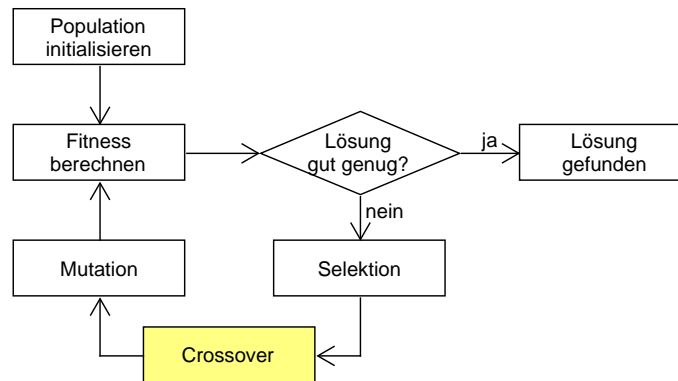


Fig. 1: Grundlegender Ablauf eines genetischen Algorithmus

Die wichtigen Parameter eines GA selbst sind zum einen die Selektion der Gene, deren Crossover-Operationen zur Erzeugung neuer Kinder, sowie die Durchführung anschließender Mutationen. Maßgeblich für die Qualität und Effizienz eines GA ist dabei die in Fig. 1 markierte Crossover-Operation.

In dieser Seminararbeit soll daher nun ein kleiner Überblick über verschiedene Crossover-Operationen gegeben werden. Nach einer Übersicht der Klassifikationen im Abschnitt 2 werden im Abschnitt 3 zuerst geeignete Anwendungen und dazugehörige Crossover-Operationen für eindimensionale, im darauf folgenden Abschnitt 4 für mehrdimensionale Repräsentationen aufgezeigt. Anschließend wird im Abschnitt 5 ein kurzer Überblick über anwendungsspezifische Codierung sowie im Abschnitt 6 ein Überblick über universell einsetzbare Crossover-Operationen gegeben.

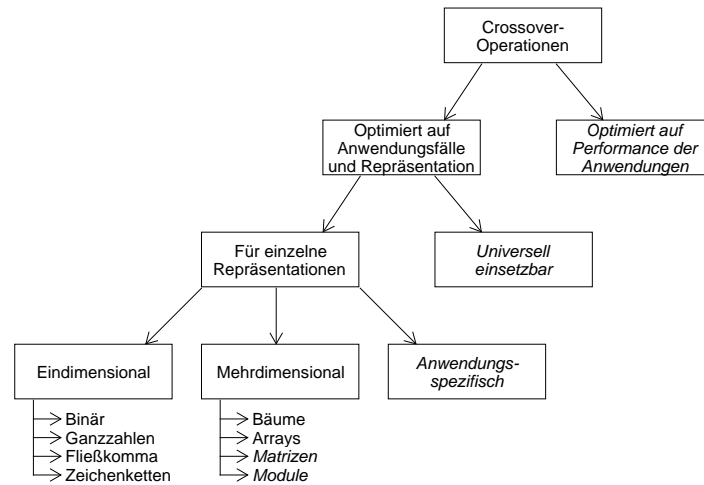


Fig. 2: Übersicht der Klassifizierung (nach [2])

2 Klassifizierungen von Crossover-Operationen

Crossover-Operationen gibt es für viele verschiedene Arten von Anwendungen und Daten. Nicht alle möglichen Operationen sind jedoch für jede Anwendung und Daten anwendbar. Pavai und Geetha haben in [2] bereits sehr gut einen Überblick über die Verschiedenen Arten von Crossover-Operationen gegeben, weshalb sich diese Seminararbeit an deren Arbeit orientiert. Crossover-Operationen können, wie in Fig. 2 dargestellt, in verschiedenen Kategorien klassifiziert werden, wobei sich diese Seminararbeit auf repräsentative Darstellungsformen der Daten und entsprechenden Crossover-Operationen beschränkt und für die kursiv markierten Klassifizierungen nur einen Überblick gibt bzw. nicht behandelt.

Die Basis aller verschiedenen Crossover-Operationen bilden die Elementaren Operationen, welche die Gene der beiden Eltern an einer oder mehreren Stellen teilen und daraus die neuen Kinder erzeugen. Entsprechend werden sie auch *One-Point-Crossover*, *Two-Point-Crossover* bzw. *N-Point-Crossover* genannt. Darauf basierend gibt es noch weitere Basis-Operationen, wie *Segmented Crossover*, bei der die Gene in eine bestimmte Anzahl Segmente anstatt an einer bestimmten Anzahl an Stellen geteilt werden [3], oder wie *Uniform Crossover*, bei der für jede Stelle des Kindes zufällig ausgewählt wird, welcher Elternteil sein Gen vererbt.

3 Eindimensionale Repräsentation

Unter eindimensionaler Repräsentation wird vor allem die Repräsentation der Daten in den elementaren Datentypen verstanden. Die Daten liegen dabei nur linear in einer bestimmten Reihenfolge vor, wodurch die Handhabung mit den

Daten auch entsprechend einfach ist und in vielen Anwendungsfällen ohne große Probleme durchgeführt werden kann und entsprechend oft genutzt wird.

Zuerst wird im Abschnitt 3.1 ein Überblick über Anwendungsfälle und mögliche Crossover-Operationen für die Binär-Codierung gegeben. Anschließend wird in Abschnitt 3.2 auf Ganzzahlige Codierung eingegangen, in Abschnitt 3.3 folgen Fließkomma-Darstellungen sowie in 3.4 Codierungen als Zeichenketten.

3.1 Binäre Codierung

Eine Codierung als Binärwerte bedeutet, dass die Werte, die vom GA bearbeitet werden, als eine Kette von 0 und 1 dargestellt sind. Der große Vorteil einer binären Codierung liegt vor allem darin, dass die Handhabung entsprechender Daten sehr einfach und platzsparend ist, weshalb diese Art der Darstellung von jeder Anwendung genutzt werden kann. Ebenfalls ein großer Vorteil liegt darin, dass entsprechende GAs aufgrund des geringen Alphabets sehr schnell und effizient sind. [1]

Vor allem folgende Arten von Anwendungen sind dafür geeignet, mit binärer Codierung zu Arbeiten:

- Klassifizierungs-Problem
- *Multimodal Spin Lattice*-Problem

3.2 Codierung als Ganzzahlen

Da ganzzahlige Werte ebenfalls sehr einfach als Binärwerte dargestellt werden können, können Crossover-Operationen für Binärwerte auch für ganzzahlige Werte eingesetzt werden. Daher wird hier vor allem auf Permutationen, also Ketten von mehreren Zahlenwerten in einer bestimmten Reihenfolge, eingegangen.

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. Herrera, F., Lozano, M., Verdegay, J.: Tackling real-coded genetic algorithms: Operators and tools for behavioural analysis. Artificial Intelligence Review 12(4), 265–319 (1998), <http://dx.doi.org/10.1023/A:1006504901164>
2. Pavai, G., Geetha, T.V.: A survey on crossover operators. ACM Comput. Surv. 49(4), 72:1–72:43 (Dec 2016), <http://doi.acm.org/10.1145/3009966>
3. Sharapov, R.R.: Genetic Algorithms: Basic Ideas, Variants and Analysis. InTech (Jun 2007), https://www.intechopen.com/books/vision_systems_segmentation_and_pattern_recognition/genetic_algorithms__basic_ideas__variants_and_analysis