

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 (CO) sind ein wesentlicher Teil von Genetischen Algorithmen (GA) und sind maßgeblich für deren Effizienz. Daher soll ein kleiner Überblick über verschiedene COs und deren Klassifizierung gegeben werden. Zunächst werden eindimensionale Repräsentationen betrachtet. Dabei werden Darstellungen als Binärwerte, Ganzzahlen bzw. entsprechende Permutationen, Fließkommazahlen und Zeichenketten erläutert, bei welchen Problemen bzw. Anwendungsfällen welche Darstellung geeignet ist, und einige der dafür optimierten COs 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 COs aufgezeigt, die nicht an eine spezielle Repräsentation der Daten gebunden sind.

1 Einführung in genetische Algorithmen

Genetische Algorithmen (GA) 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 evaluiert werden und anschließend nur die besten in der Population verweilen dürfen.

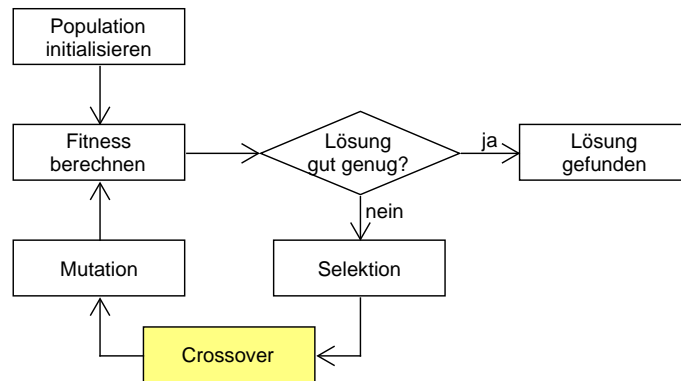


Fig. 1: Grundlegender Ablauf eines genetischen Algorithmus

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

In dieser Seminararbeit soll daher nun ein kleiner Überblick über einige verschiedene COs gegeben werden. Nach einer Übersicht der Klassifikationen im Abschnitt 2 werden im Abschnitt 3 zuerst geeignete Anwendungen und dazugehörige COs 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 COs gegeben.

2 Klassifizierungen von Crossover-Operationen

COs gibt es für viele verschiedene Arten von Anwendungen und Daten. Nicht alle möglichen Operationen sind jedoch für jede Anwendung und Daten anwendbar.

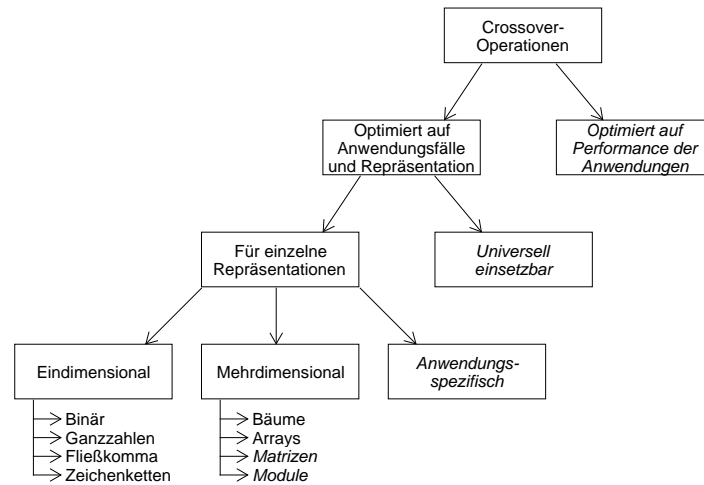


Fig. 2: Übersicht der Klassifizierung (nach [11])

Pavai und Geetha haben in [11] bereits sehr gut einen Überblick über die verschiedenen Arten von COs gegeben, weshalb sich diese Seminararbeit an deren Arbeit orientiert. COs können, wie in Fig. 2 dargestellt, in verschiedenen Kategorien klassifiziert werden. Diese Arbeit beschränkt sich auf die repräsentativen Darstellungsformen der Daten und deren entsprechenden COs. Für die kursiv markierten Klassifizierungen wird daher nur ein kurzer Überblick bzw. im Falle der performance-optimierten COs darauf verzichtet.

Die Basis aller verschiedenen COs 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 *N-Point-Crossover* bzw. im Speziellen auch *One-Point-Crossover* oder *Two-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 [12], 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 Darstellung 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 COs für die Binär-Codierung gegeben. Anschließend wird in Abschnitt 3.2 auf ganzzahlige Codierung allgemein bzw. auf Permutationen im Speziellen eingegangen, in Abschnitt 3.3 folgen Fließkomma-Darstellungen sowie in 3.4 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 prinzipiell von jeder Anwendung genutzt werden kann. Ebenfalls ein großer Vorteil liegt darin, dass entsprechende GAs aufgrund des geringen Alphabets sehr schnell und effizient sind. [4]

Vor allem folgende Arten von Anwendungen sind besonders dafür geeignet, mit binärer Codierung zu Arbeiten: [11]

Classification Problem Lösung soll in verschiedene Kategorien klassifiziert werden [5]

Multimodal Spin Lattice Problem Suchen eines minimalen Energiezustandes für 450 Spins mit je vier Zuständen auf einem 2D-Gitter [9]

Passende Crossover-Operationen dafür sind:

Self-Crossover Tauschen von einzelnen Bits innerhalb des Chromosoms [10]

Supplementary Crossover Nutzt das *Center of Gravity*-Paradigma um Kinder zu erzeugen [2]

Generalized crossover Interpretiert Chromosom als Ganzzahl und dividiert diese durch eine andere, zufällig ausgewählte Ganzzahl [3]

3.2 Codierung als Ganzzahlen und Permutationen

Da ganzzahlige Werte ebenfalls sehr einfach als Binärwerte dargestellt werden können, können COs für Binärwerte prinzipiell auch für ganzzahlige Werte eingesetzt werden. Natürlich gibt es auch entsprechende Anwendungsfälle und COs, die speziell für ganzzahlige Darstellungen geeignet und optimiert sind, auf die hier aber verzichtet wird und stattdessen auf Ketten von ganzzahligen Werten, also Permutationen, eingegangen wird.

Der Unterschied zwischen einfacher ganzzahligen Darstellungen und Permutationen liegt darin, dass bei Permutationen die komplette Zahlenfolge als mehrere aneinandergereihte Zahlen betrachtet werden muss. Entsprechend können nicht einfache COs für Binärwerte und einfache ganzzahlige Werte eingesetzt werden, sondern COs, welche die Zahlenketten entsprechend berücksichtigen. Dies ist u. A. bei folgenden Anwendungen und Problemen der Fall:

Traveling Salesman Problem (TSP) Mehrere Orte, die mit einer möglichst kurzen Strecke miteinander verbunden werden müssen [1]

Graph Coloring Problem Knoten eines Graphen mit möglichst wenig und verschiedenen Farben einfärben [8]

Quadratic Assignment Problem Summe der Distanzen zwischen Punkten minimieren, ähnlich wie TSP, nur mit quadratischer Kostenfunktion [7]

Passende COs für Permutationen lassen sich in verschiedene Kategorien einteilen, welche sich basierend auf dem grundlegenden Vorgehen der COs klassifizieren lassen. Im Folgenden werden nun einige positionsbasierte, kantenbasierte und folgenbasierte COs vorgestellt. Daneben gibt es aber auch noch weitere, wie z. B. teilmengenbasierte, die auf Basis von Genom-Teilmengen arbeiten, *Cut and Splice*-basierte, die einen Teil der Elterngenome tauschen, oder distanzbasierte COs, bei denen die Anzahl der unterschiedlichen Gene zwischen allen Elternteile gleich sein muss.

Positionsbasierte COs In dieser Kategorie finden sich vor allem COs, welche positionsbasiert einzelne Gene beeinflussen. Dies sind u. A. folgende:

Partially Mapped Crossover (PMX) Wählt zufällig zwei Punkte aus und tauscht alle Werte dazwischen mit dem anderen Elternteil. Bei der Erweiterung Uniform PMX werden einzelne Gene anstatt eines ganzen Segments getauscht. [1] [6] [7]

Position Based Crossover (POS) Wählt einige zufällige Positionen im ersten Elternteil aus und verschiebt die ausgewählten Werte zu den korrespondierenden Positionen im anderen Elternteil. [6]

Kantenbasierte COs In dieser Kategorie finden sich COs, welche basierend auf den Kanten der Eltern(-knoten) neue Kinder erzeugen. Darunter fallen u. A. folgende COs:

Edge Crossover Wählt einen Knoten mit der geringsten Kantenzahl aus und fügt ihn zum neuen Kind hinzu und löscht ihn aus den anderen Kantenlisten. [6]

Edge Exchange Crossover (EXX) Vererbt zunächst alle Elternkanten um einen Kreis zu bilden, entfernt dann aber ungültige Kanten. Falls kein Kreis gebildet werden konnte, werden alle Kanten vererbt. [13]

Folgenbasierte COs Folgenbasierte COs beeinflussen vor allem die Reihenfolge der Gene. Sie lassen sich in weitere Unterkategorien einteilen. Darunter fallen u. A. folgende COs:

Merging Crossover (MOX) Beide Eltern werden zuerst zufällig zusammengefügt und anschließend in die beiden Kinder geteilt. [8]

Non-Wrapping Ordered Crossover Erstellt Lücken und füllt diese wieder auf, ohne dabei die absolute Reihenfolge der Gene zu verlieren. [1]

Daneben gibt es auch noch weitere COs, welche sich neben den hier vorgestellten Unterkategorien (verschmelzende und absolute Reihenfolge) z. B. in sortierende oder angrenzende, folgenbasierte COs einsortieren lassen.

3.3 Codierung als Fließkommazahl

Im Gegensatz zur Codierung als Ganzzahlen bzw. Binärwerte lassen sich Daten, welche als Fließkommazahlen codiert sind, viel genauer darstellen, haben dadurch aber auch einen deutlich größeren Suchraum für mögliche Lösungen. Es gibt aber auch viele Probleme und Anwendungsfälle, bei denen sich die Darstellung als Fließkommazahl deutlich besser eignet. Dies liegt auch dadurch begründet, dass viele Problemfälle bereits durch die Codierung nicht sehr effizient bearbeitet werden können, da die Auflösung als Binär- bzw. ganzzahlige Werte zu gering ist. Ebenso entfällt bei einer Fließkomma-Codierung und der Nutzung entsprechender COs der Bedarf der Konvertierung der Daten in eine andere Darstellungsform, wodurch letztlich die Effizienz des GAs deutlich erhöht wird.

Vor allem Anwendungen, welche einen kontinuierlichen Suchraum besitzen, lassen sich mit Fließkomma-Codierungen effizient ausführen. Darunter fallen z. B. folgende Anwendungen und Probleme:

Animal Diet Formulation Problem Erstellung eines Ernährungsplans mit möglichst vielen Nährstoffen zu minimalen Kosten

Electromagnetic Optimization Handhabung mit elektromagnetischen Werten in einem kontinuierlichem Suchraum

Revenue Management in Airlines Zuweisen von begrenzten Ressourcen einer Airline (Flugzeuge, Personal, ...), um den Gewinn zu maximieren

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. Abdoun, O., Abouchabaka, J.: A comparative study of adaptive crossover operators for genetic algorithms to resolve the traveling salesman problem. CoRR abs/1203.3097 (2012), <http://arxiv.org/abs/1203.3097>
2. Angelov, P.: Supplementary crossover operator for genetic algorithms based on the center-of-gravity paradigm. *Control and Cybernetics* Vol. 30, no 2, 159–176 (2001)
3. Coli, M., Gennuso, G., Palazzari, P.: A new crossover operator for genetic algorithms. In: *Proceedings of IEEE International Conference on Evolutionary Computation*. pp. 201–206 (May 1996)
4. 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>
5. Ho, S.Y., Liu, C.C., Liu, S.: Design of an optimal nearest neighbor classifier using an intelligent genetic algorithm. *Pattern Recognition Letters* 23(13), 1495 – 1503 (2002), <http://www.sciencedirect.com/science/article/pii/S0167865502001095>
6. Kumar, R., Gopal, G., Kumar, R.: Novel crossover operator for genetic algorithm for permutation problems 3 (May 2013)
7. Misevičius, A., Kilda, B.: Comparison of crossover operators for the quadratic assignment problem. In: *Information technology and control*. vol. 34 (2005)
8. Mumford, C.L.: New Order-Based Crossovers for the Graph Coloring Problem, pp. 880–889. Springer Berlin Heidelberg, Berlin, Heidelberg (2006), http://dx.doi.org/10.1007/11844297_89
9. Pál, K.F.: Selection schemes with spatial isolation for genetic optimization, pp. 169–179. Springer Berlin Heidelberg, Berlin, Heidelberg (1994), http://dx.doi.org/10.1007/3-540-58484-6_261
10. Pal, N.R., Kundu, M.K., Nandi, S.: Application of a new genetic operator in feature selection problems. In: *TENCON '98. 1998 IEEE Region 10 International Conference on Global Connectivity in Energy, Computer, Communication and Control*. vol. 1, pp. 37–40 vol.1 (1998)
11. 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>
12. 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
13. Takahashi, R.: A methodology of extended changing crossover operators to solve the traveling salesman problem. In: *2008 Fourth International Conference on Natural Computation*. vol. 1, pp. 263–269 (Oct 2008)