

KI-gestützte Optimierung repetitiver Prozesse

Eine Kodierungstechnik für repetitive Prozesse in der evolutionären Optimierung

Christina Plump, Rolf Drechsler, Universität Bremen und DFKI- Cyber-Physical Systems und Bernhard J. Berger, Technische Universität Hamburg (TUHH)

Optimierung ist eine wesentliche Fragestellung in vielen Kontexten. Sei es Ressourcen-, Zeit-, Personal- oder nicht zuletzt auch Kosteneffizienz, regelmäßig müssen Prozesse, Einstellungen, Zusammensetzungen – die Liste ließe sich beliebig fortsetzen – optimiert werden. Um das Optimierungsproblem zu lösen, gibt es viele unterschiedliche Techniken. Eine besondere Klasse stellen die evolutionären Algorithmen dar, sie zählen zu den populationsbasierten, heuristischen Verfahren. Sie erlauben auch die Optimierung von Problemen mit vielen lokalen Optima oder stark durch Nebenbedingungen eingeschränkten Suchräumen. Gleichzeitig sind sie in der Lage, im Rahmen eines einzelnen Optimierungslaufs mehrere Lösungsmöglichkeiten vorzuschlagen. Ein besonderer Aspekt bei der Verwendung von evolutionären Algorithmen ist die Wahl der korrekten Kodierung oder die wohldefinierte Spezifikation einer Kodierung. Insbesondere bei der Optimierung eines Prozesses, der aus sich wiederholenden Grundprozessen besteht, die sich nur in ihren Parameterausprägungen unterscheiden, ist dies ein nicht-triviales Problem. Dieser Beitrag stellt eine solche, bereits erfolgreich getestete Kodierung vor, die sich insbesondere in Hinblick auf die KI-Unterstützung als dateneffizient herausgestellt hat.

Evolutionäre Algorithmen sind eine populationsbasierte, heuristische Technik zur Optimierung. Sie basieren auf dem biologischen Prinzip der Evolution. Das grundlegende Prinzip ist in Bild 1 zu sehen. Eine Population aus Lösungskandidaten wird hinsichtlich der Zielfunktion bewertet – man sagt, ihnen wird eine „Fitness“ zugewiesen. Die Fitnessfunktion ist also schlicht die Auswertung der Zielfunktion des Optimierungsproblems. Ein gewisser Prozentsatz der Population wird anhand dieser Fitness zur Reproduktion ausgewählt – dies wird auch Selektion genannt. Es folgt die Rekombination von Elternindividuen aus denen Kindindividuen entstehen. Anschließend werden zufällig ausgewählte Individuen (aus Eltern- oder Kindpopulation) mutiert. Dies bildet die Population der nächsten Generation und der Prozess beginnt von neuem. Der Vorgang endet, wenn entweder das Optimierungsziel in der Population erreicht ist, die Anzahl der erlaubten Generationen erreicht ist oder die Popula-

tion sich in einem Maße assimiliert hat, dass keine Fortschritte durch Rekombination mehr zu erwarten sind. Um die Schritte der Rekombination und Mutation durchzuführen, ist eine Kodierung der Lösungskandidaten notwendig. Das kann eine Kodierung in Bitform sein, aber auch die schlicht reellwertige Kodierung von numerischen Problemen ist häufig zu sehen. Hierbei wird die Kodierung des Lösungskandidaten auch als Genotyp (in Anlehnung an die Biologie) bezeichnet, während die einzelnen Dimensionen des Lösungskandidaten (bei mehrdimensionalen Suchräumen) als Chromosome bezeichnet werden. Jedes Chromosom wird dann auf eine bestimmte Art und Weise kodiert, zum Beispiel eben durch eine Bitrepräsentation oder eine reellwertige Darstellung.

Evolutionäre Algorithmen (EA)

Für evolutionäre Algorithmen ist der entscheidende Punkt die Wahl der entsprechenden Komponenten (Kodie-

AI-Supported Optimization of Repetitive Processes

Optimisation is an essential task in many situations. The class of evolutionary algorithms is a population-based, heuristic technique for optimisation. They allow the optimisation of multi-modal problems even with distorted search spaces. They can propose several solutions instead of just one. An important aspect of evolutionary algorithms is encoding search space candidates. In the optimisation of processes, this is a non-trivial task. This article describes a successfully tested encoding.

Keywords:

artificial intelligence, machine learning, evolutionary algorithm, optimization, encoding



Dipl.-Math., Dipl.-Inf. Christina Plump arbeitet als Researcher am DFKI Bremen, Abteilung Cyber-Physical Systems mit dem Schwerpunkt Optimierung unter Unsicherheit.



Prof. Dr. Rolf Drechsler leitet die Arbeitsgruppe Rechnerarchitektur der Universität und ist Abteilungsleiter des Gebiets „Cyber-Physical Systems“ des DFKI.



Dr. Bernhard J. Berger arbeitet als Oberingenieur im Institut für eingebettete Systeme an der Technischen Universität Hamburg (TUHH).

cplump@uni-bremen.de
www-cps.hb.dfk.de

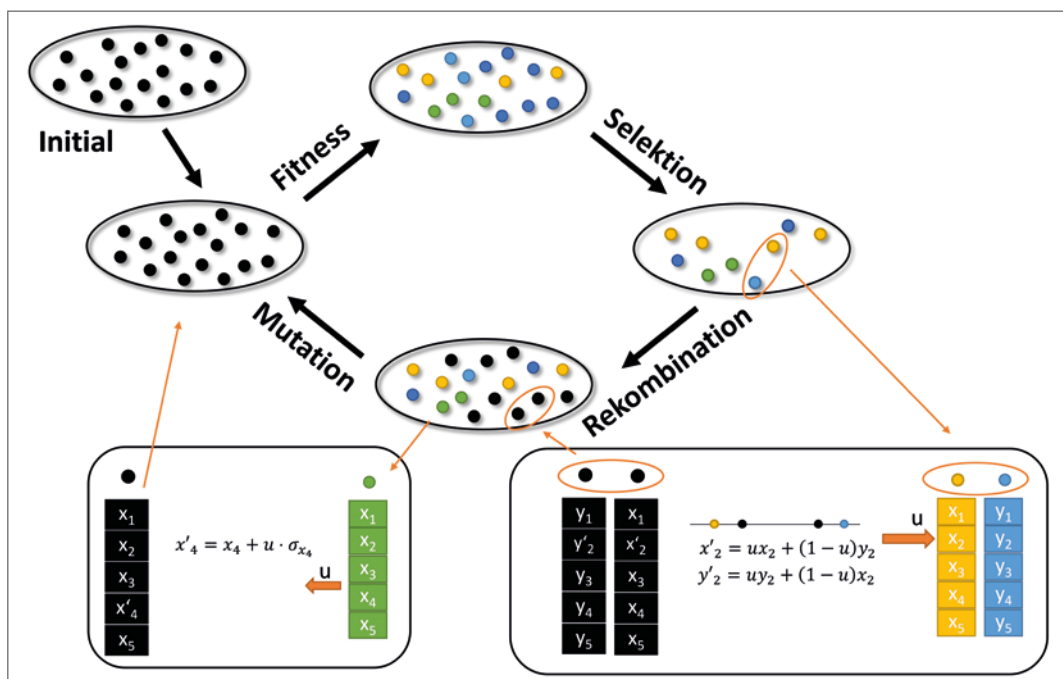


Bild 1: Schematische Darstellung des Ablaufs eines evolutionären Algorithmus, sowie beispielhafter Mutations- und Rekombinationsoperatoren.

ung, Selektion, Rekombination und Mutation). Sie alle müssen zueinander passen und gleichzeitig zur Problemstellung (Zielfunktion und Suchraum). [1] gibt eine gut verständliche Einführung in die Thematik.

Ist die Zielfunktion der Optimierung unbekannt oder nur kostenintensiv zu berechnen (beispielsweise durch Experimente oder das Lösen komplexer Gleichungen), können approximierte Zielfunktionen gute Dienste leisten. Hierfür werden zunächst in einer Vorbereitungsphase Trainingsdaten gesammelt und dann wird mithilfe eines geeigneten Verfahrens des maschinellen Lernens, z. B. neuronale Netze, die Zielfunktion geschätzt. Im Optimierungsprozess kann dann diese Schätzung verwendet werden, die kostengünstiger ist. Natürlich hat die Nutzung der geschätzten Funktion ihre Nachteile, insbesondere in Bereichen des Suchraums, für die die Schätzung weniger gut gelingt. Hier gibt es allerdings Techniken wie zum Beispiel die Einbeziehung der Verteilung der Trainingsdaten, sodass der negative Einfluss der Schätzung auf den Erfolg der Optimierung möglichst geringgehalten wird. Diese Notwendigkeit erfordert eine Form der Kodierung, die es ermöglicht, diese geschätzte Zielfunktion anzuwenden. Einen Überblick über die Thematik der approximierten Fitnessfunktionen findet sich in [2].

Sei in Kürze ein Standardbeispiel betrachtet (zum besseren Verständnis): Dieser Fall ist eine Kostenfunktion, die minimiert werden soll mit fünf Eingabewariablen, die alle numerisch sind. Hier ist eine Kodierung sinnig, die die Identität darstellt, d. h. ein Lösungskandidat wird als Tupel

seiner Werte dargestellt (x_1, x_2, x_3, x_4, x_5). Als Rekombination ließe sich gut ein Operator wählen, der zufällig einen auf der Verbindungslinie beider Eltern liegenden Punkt wählt. Soll der Suchraum explorativ erforscht werden, so können dies „außerhalb“ der Eltern liegende Punkte sein, soll der Suchraum feingranular erforscht werden, ließe sich der Operator auf die innenliegenden Punkte beschränken. Ein Mutationsoperator in diesem Fall könnte die Addition eines Vielfachen der Standardabweichung der zur Mutation ausgewählten Variable sein (Bild 1). Die Wahl der Operatoren und ihrer Ausprägung (Parameterwahl) ist entscheidend für die Fähigkeit des Algorithmus, lokale Optima zu verlassen, um das globale Optimum zu erreichen.

Der folgende Abschnitt erläutert die Herkunft und Herausforderung von repetitiven Prozessen, d.h. Prozessen, die aus sich wiederholenden Grundprozessen bestehen, die die zugrunde liegende Forschung motiviert haben.

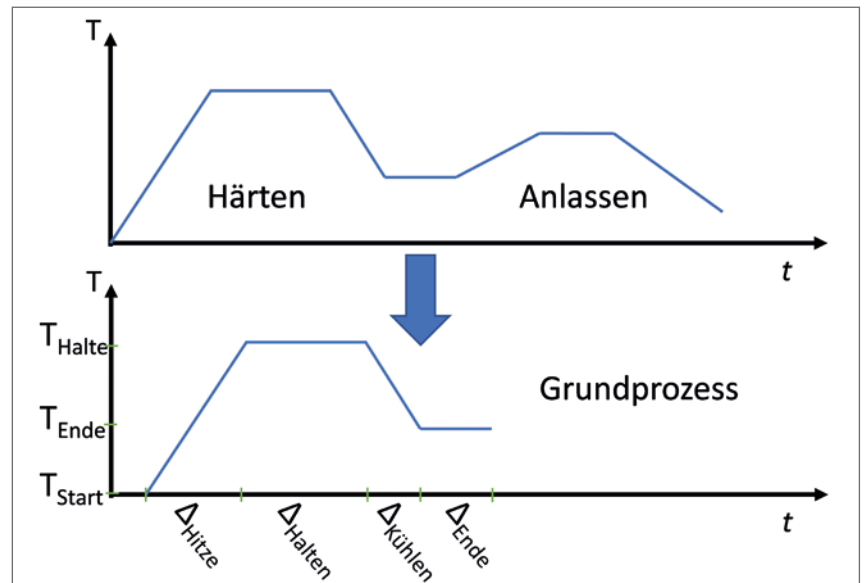
Repetitive Prozesse

Prozesse sind ein essenzieller Bestandteil vieler Unternehmungen. Sie kommen in der Logistik vor, in der Herstellung, in der Forschung – auch die Wertschöpfungskette selbst ist ein Prozess. Entsprechend häufig geht es eigentlich um eine Optimierung des gesamten Prozesses und nicht nur einzelner Parameter innerhalb dieses Prozesses. In der Methode „Farbige Zustände“ [3, 4] beispielsweise lag ein besonderes Augenmerk auf dem Prozess der Wärmebehandlung von Konstruktionswerkstoffen. Es galt diesen dahingehend zu optimieren, dass der resultierende

Konstruktionswerkstoff genau die Eigenschaften besaß, die durch ein vorher spezifiziertes Anforderungsprofil vorgegeben waren. Diese Eigenschaften waren beispielweise die Härte des resultierenden Werkstoffs, die Druckfestigkeit, die Zugfestigkeit und andere Werkstoffeigenschaften. Eine typische Wärmebehandlung besteht aus einer starken Erwärmung (auch Austenitierung genannt) und nach einem Abschrecken einer nachgelagerten (meist) schwächeren Erwärmung, die auch als Anlassen bezeichnet wird (bzw. Vergüten, wenn die Temperatur signifikant höher ist) (Bild 2 für eine schematische Darstellung). Allerdings gibt es auch Stähle, die einen dritten Erwärmungsschritt benötigen, um die gewünschten Eigenschaften zu erhalten. Auf der Suche nach neuartigen Konstruktionswerkstoffen ist es prinzipiell auch nicht ausgeschlossen, dass irgendwann weitere dieser Erwärmungszyklen hinzukommen. Prinzipiell handelt es sich also um einen repetitiven Prozess, der im Prinzip beliebig oft wiederholt werden kann mit potenziell veränderlichen Parametern. Dasselbe gedankliche Prinzip ließe sich auf viele Prozesse übertragen, die in einer variablen Anzahl wiederholt werden können und dabei in ihren einzelnen Parametern unterschiedliche Werte annehmen können.

Soll über diese Prozesse nun eine Optimierung stattfinden und insbesondere die Variabilität in der Anzahl der potenziell modifizierten Wiederholungen beibehalten werden, so tritt automatisch die Herausforderung zu Tage, dass diese Prozesse in irgendeiner Form für den evolutionären Algorithmus passend kodiert werden müssen. Eine schlichte Variante wäre die Form (im Fall der Wärmebehandlungsprozesse), die Eckpunkte auf dem Temperaturdiagramm zu kodieren, in schlichter abwechselnder Folge der x - und y -Werte. Dies hat jedoch den Nachteil, dass die Rekombination und Mutation des evolutionären Algorithmus nicht nur realitätsferne, sondern schlicht nicht darstellbare Prozesse erzeugen würden. Nicht mal mehr der Anspruch einer Temperatur-Zeit-Kurve wäre in dem Sinne zu halten, insbesondere weil keine Ordnung auf den x -Werten (für die Zeit) eingehalten werden könnte. Dies gilt in dieser Form auch für andere zu kodierende Prozesse.

Durch die hierarchische Struktur der Prozesse – ein Satz an Parametern gehört zu einem Prozess, mehrere unterschiedliche Instanzen dieser Prozesse werden wiederholt – und die gleichzeitig bisher eindimensionale Struktur der Kodierung für evolutionäre Algorithmen entsteht stets die Problematik, dass durch die Mutation und Rekombination die Grundstruktur der Prozesse zerstört werden könnte.



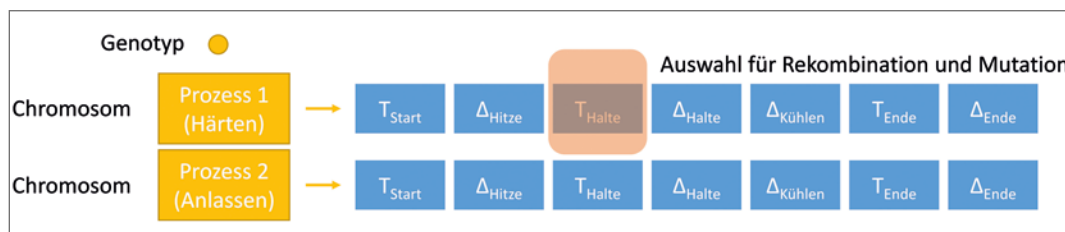
Kodierung von repetitiven Prozessen

Wie kann nun mit diesen repetitiven Prozessen in Bezug auf die Kodierung für evolutionäre Algorithmen umgegangen werden? Auch und insbesondere im Hinblick auf die Tatsache, dass die zugehörige Fitnessfunktion möglicherweise durch maschinelles Lernen approximiert werden muss? Im vorigen Absatz hat sich bereits angedeutet, dass die eindimensionale Struktur der Kodierung problematisch ist, sowie die Anpassung der Operatoren auf die Kodierung, damit diese stets valide (nicht sinnvolle, aber valide) Prozesse herstellen.

Die Lösung ist der Gang in die Zweidimensionalität und die Auflösung der alten Logik: Ein Chromosom für eine Dimension. Eine Folge repetitiver Prozesse wird nun als Tupel seiner Grundprozesse dargestellt. Das heißt, ein einzelnes Chromosom stellt nun nicht mehr eine Dimension des Suchraums dar, sondern einen Grundprozess. Was in der Folge die Frage aufwirft, wie ein solcher (einzelner) Grundprozess (siehe einen Erwärmungsvorgang im Beispiel) dazustellen ist. Hier wurde sich für die Markierung der charakteristischen Punkte des Erwärmungsprozesses entschieden – im Kontext des Projekts – d. h. es werden die Starttemperatur, Erwärmungsdauer, Zieltemperatur, Haltedauer, Abkühldauer und Endtemperatur eines Prozesses kodiert. Bild 3 zeigt hier schematisch, wie diese Kodierung beispielhaft für die Temperaturkurve aus Bild 2 anzunehmen ist. Die Kodierung des zugrundeliegenden Prozesses jedoch ist grundsätzlich an die jeweilige Domäne anzupassen. Hier könnten auch Rüstzeiten, Produktionszeiten und Lagerzeiten stehen, bevor der Prozess an einer weiteren Station wiederholt wird (mehrere Produktionsvorgänge an einem Produkt). Die Variabilität der Anzahl der Repetitionen der

Bild 2: Wärmebehandlungsprozesse und der daraus abzuleitende Grundprozess.

Bild 3: Schematische Darstellung der Kodierung am Beispiel der Temperaturkurve.



Prozesse kann durch die Einführung einer Maximalzahl geregelt werden. Diese ist so zu wählen, dass mehr als diese Wiederholungen entweder technisch nicht möglich oder in anderer Form realistisch nicht sinnig wären. Dann wird in der Kodierung stets diese Anzahl an Chromosomen freigehalten. Durch Mutation und Rekombination können sich Prozesse dann verlängern und verkürzen.

Die Rekombination und Mutation hat nun zwei Aufgaben – zum einen die Modifikation der einzelnen Prozesse, zum anderen die Verkürzung oder Verlängerung der gesamten Folge von Prozessen. Die Rekombination nimmt zwei Grundprozesse als Eingabe und produziert zwei Grundprozesse nach dem oben erklärten Prinzip: Es werden zufällig Parameter der Grundprozesse ausgewählt, bspw. Erwärmungsdauer und Endtemperatur. Für diese werden dann jeweils zwei Punkte auf der sie verbindenden Linie gewählt. Jeder Grundprozess enthält so eine neue Kombination der Parameter Erwärmungsdauer und Endtemperatur. So ist sichergestellt, dass beide resultierenden Grundprozesse auch valide Prozesse sind und die gesamte Prozessfolge somit ebenfalls. Für die Mutation eines Grundprozesses werden zufällig Parameter des Prozesses gewählt, die passend zum obigen Beispiel dann entsprechend einer einstellbaren Standardabweichung modifiziert werden.

Tatsächlich ändern sich Rekombination und Mutation kaum, sie verschieben sich nur von der Ebene des Genotyps, indem Chromosome rekombiniert und mutiert wurden auf die Ebene des Chromosoms, auf der nun Parameter rekombiniert und modifiziert werden. Die Mutation kann so auch Prozesse verlängern: Wird ein Grundprozess (also ein Chromosom) zur Modifikation ausgewählt, der noch keine Parameter hat (bzw. diese alle bei 0 liegen), kann durch die Mutation einzelner Parameter wie oben beschrieben ein tatsächlicher Grundprozess erzeugt und somit die Prozessfolge verlängert werden.

Die Form dieser Kodierung hat zusätzlich einen entscheidenden Vorteil für den Nutzen des maschinellen Lernens in Bezug auf die Fitnessfunktion. In einer eindimensionalen Darstellung wäre es notwendig, Trainingsdaten für beliebige Längen

der Prozessfolgen sowie Grundprozesse zu erheben, um eine Fitnessfunktion approximieren zu können. Betrachtet man die Grundprozesse als zeitliche Dimensionen, so können sie als Operation auf dem aktuellen Zustand des Systems (im Fall der Wärmebehandlung, des Konstruktionswerkstoffs) verstanden werden. Das bedeutet, dass zur Berechnung der Fitness eines Lösungskandidaten nicht der gesamte Prozess auf einmal auf den ursprünglichen Zustand angewendet wird, sondern die Teilprozesse Stück für Stück. Im realen Verfahren wird der Gesamtprozess durchgeführt, aber die Berechnung findet nur Schritt für Schritt statt. Eine Prozessfolge würde so Stück für Stück auf den aktuellen Systemzustand angewandt und nach Ermittlung des letzten Zustands, die Fitness desselbigen geschätzt. Diese Technik ermöglicht es, den Suchraum für die Trainingsdaten zu verringern. Nun müssen nur unterschiedliche Prozesse auf unterschiedlichen Grundzuständen des Systems als Trainingsdaten aufgefasst werden, da pro Abbildung nur ein Prozess auf einen gegebenen Zustand abgebildet wird.

Auf das motivierende Problem angewandt führte dies bereits bei 80 unterschiedlichen eingestellten Zuständen zu einer guten Erfolgsquote bei der Suche nach Wärmebehandlungsprozessen, die genau die Zustände im Konstruktionswerkstoff einstellten, die gewünscht waren [5].

Alles in allem hat die Erweiterung des Kodierungskonzepts auf zwei Dimensionen es ermöglicht, Prozessfolgen schlüssig darzustellen und insbesondere sicherzustellen, dass die Konzepte des evolutionären Algorithmus während der Optimierung ihre Fähigkeiten entfalten können. Zusätzlich ermöglicht diese Form der Darstellung ein dateneffizientes maschinelles Lernen, was insbesondere bei experimentell oder in realer Umgebung erhobenen Trainingsdaten von Vorteil ist.

Die Forschung zu diesem Beitrag wurde durch das Projekt „Farbige Zustände“ (SFB1232), das von der Deutschen Forschungsgemeinschaft gefördert wurde, inspiriert.

Schlüsselwörter: Künstliche Intelligenz, Maschinelles Lernen, Evolutionäre Algorithmen, Optimierung, repetitive Prozesse, Kodierung, Rekombination, Mutation

Literatur

[1] Weicker, K.: Evolutionäre Algorithmen. Berlin Heidelberg 2015.
 [2] Jin, Y.: Surrogate-assisted evolutionary computation: Recent advances and future challenges. In: Swarm and Evolutionary Computation 1 (2011) 2, S. 61-70. <https://doi.org/10.1016/j.swevo.2011.05.001>.
 [3] Ellendt, N.; Mädler, L.: High-Throughput Exploration of Evolutionary Structural Materials. In: HTM Journal of Heat Treatment and Materials 73 (2018), S. 3-12. [10.3139/105.110345](https://doi.org/10.3139/105.110345).
 [4] Drechsler, R.; Eggersgläub, S.; Ellendt, N.; Huhn, S.; Mädler, L.: Exploring superior structural materials using multi-objective optimization and formal techniques 2016. In: Sixth International Symposium on Embedded Computing and System Design (ISED), Patna, India, (2016), S. 13-17, doi: 10.1109/ISED.2016.7977046.
 [5] Ellendt, N. u. a.: Experimental Methods to Enable High-Throughput Characterisation of New Structural Materials. In: JOM (2021).