

GENÉTICO ALGORITMO PARA A SELECÇÃO DE ATLETAS: O EFEITO MONEYBALL

BARRETO, Ana Cristina Lopes y Glória¹; BRASIL, Roxana Macedo²; CARVALHO JUNIOR, Sergio³; BRITO, Diogo de Freitas^{1,4}; JUNIOR, Homero da Silva Nahum^{1,5}

Resumo

O trabalho objetivou desenvolver um modelo de Algoritmo Genético (AG) para seleccionar o maior número de jogadores de voleibol dentre 10 candidatos. Para cada atleta foi criado um índice de Custo e Ponto, então o modelo deveria maximizar o Ponto, porém com Custo < 53. Esse problema tinha 1009 combinações possíveis. O AG do tipo binário seleccionou oito atletas, ao Custo = 49 e conquistou Pontos = 52. Possível foi concluir que o objetivo foi conquistado adequadamente.

Palavras-chave: Meta-heurística. Algoritmo evolutivo. Inteligência artificial. Computação científica. Ciências de dados.

Zusammenfassung

Ziel dieser Arbeit war es, ein Modell des Genetischen Algorithmus (GA) zu entwickeln, das aus 10 Kandidaten die größte Anzahl von Volleyballspielern auswählt. Für jeden Athleten wurde ein Kosten- und Punkteindex erstellt, so dass das Modell die Punkte maximieren musste, aber mit Kosten < 53. Dieses Problem hatte 1009 mögliche Kombinationen. Die binäre GA wählte acht Spieler zu Kosten = 49 aus und erhielt Punkte = 52. Es konnte festgestellt werden, dass das Ziel angemessen erreicht wurde.

Schlüsselwörter: Meta-Heuristiken. Evolutionärer Algorithmus. Künstliche Intelligenz. Wissenschaftliches Rechnen. Datenwissenschaft.

Abstract

The aim of this work was to develop a Genetic Algorithm (GA) model to select the greatest number of volleyball players out of 10 candidates. A Cost and Point index was created for each player, so the model had to maximise the Point, but with a Cost < 53. This problem had 1009 possible combinations. The binary GA selected eight players at Cost = 49 and obtained Points = 52. It was possible to conclude that the objective was achieved adequately.

Keywords: Meta-heuristics. Evolutionary algorithm. Artificial intelligence. Scientific computing. Data science.

¹ Docentes do Curso de Educação Física do Centro Universitário Celso Lisboa;

² Docente Ph.D. em Educação Física;

³ Pesquisador convidado do BIODESA;

⁴ Docente do Curso de Gestão Desportiva e do Lazer do Centro Universitário Celso Lisboa;

⁵ Docente da Escola de Saúde da Universidade Cândido Mendes.

Einleitung

Der Genetische Algorithmus (GA) ist eine Meta-Heuristik, die auf dem Prozess der natürlichen Selektion basiert (Linden, 2012), was bedeutet, dass das am besten an die Umweltbedingungen angepasste Individuum die größten Überlebenschancen hat. Es handelt sich also um einen evolutionären Algorithmus, der die biologische Evolution in die Mathematik überträgt, um die Lösung eines bestimmten Problems zu finden (Sáez-Gutiérrez *et al.*, 2024), und der 1992 von Holland vorgeschlagen wurde (Katoch, Chauhan und Kumar, 2021).

Die konzeptionelle Konsequenz daraus war, dass die GA eine lokale Suchmaschine wäre, so dass die Lösung würde in der Nähe des aktuellen Zustands gesucht werden. Pragmatisch gesehen würde die Suche jedoch im gesamten verfügbaren Raum stattfinden, was sie als globale Suche klassifizieren würde (Russel und Norvig, 2022). Infolgedessen besteht die Herausforderung für GAs darin, hochkomplexe und konkurrierende Ziele auszugleichen (Pinto, Martarelli und Nagano, 2022; Michalewicz, 2013):

- *Exploration* (Erkundung): Neue Regionen im Suchraum zu untersuchen;
- *Exploitation* (Intensivierung): Den größten Nutzen aus dem erworbenen Wissen zu ziehen und die besten Lösungen an den vielversprechendsten Punkten des bereits besuchten Suchraums zu finden.

Die Anwendung dieser Idee würde das Verständnis bestimmter Konzepte voraussetzen. Zunächst muss man sich darüber im Klaren sein, dass jede mögliche Lösung für ein bestimmtes Problem alle gemessenen Variablen einschließt. In der GA würde dies bedeuten, dass jedes Individuum aus allen beteiligten Genen besteht, was auf die folgenden grundlegenden Arten geschehen kann (Eiben und Smith, 2015):

- Binär: Die Individuen wären Folgen von Bits (0 und 1). Diese Darstellung eignet sich für kombinatorische Optimierungsprobleme, die darauf abzielen, Elemente unter bestimmten Bedingungen auszuwählen. Beim Kauf von bis zu sechs Elementen (Genen) mit einer bestimmten Höchstmenge wäre ein mögliches Individuum 010101, was bedeutet, dass die Elemente an den ungeraden Positionen nicht ausgewählt werden;
- Permutation: Die Individuen wären numerische Sequenzen und eignen sich daher für Probleme, die eine bestimmte Reihenfolge erfordern, wie sie bei der Routenplanung oder der Terminplanung häufig vorkommen. Ein Beispiel: Ein Lastwagen muss acht Städte beliefern, wobei er sie nur einmal durchfährt.

Jede Stadt wäre ein Gen, wenn die Nummerierung 1, 2, 3, 4, 5, 6, 7 und 8 lautet, dann wäre ein Individuum 61243758;

- Realer Wert: Wenn das Ziel darin besteht, Funktionen mit kontinuierlichen Variablen zu optimieren, dann wäre das Individuum ein Vektor aus reellen Zahlen. Bei einer Funktion mit vier Variablen könnte das Individuum zum Beispiel (9.4, 0, -3.7, 6) sein.

Folglich würde die Menge aller möglichen Lösungen (Individuen) als Population bezeichnet werden. Da es sich um eine Meta-Heuristik handelt, muss die Ausgangspopulation generiert werden (Michalewicz und Fogel, 2013), was durch die Verwendung von Seeds, einer Zufallszahl, die zum Starten eines Algorithmus verwendet wird (L'Ecuyer, 1994; Park und Miller, 1998), geschehen kann, die in zufälligen Individuen gipfelt, die der Bewertungsfunktion vorgelegt werden. Die Individuen mit den höchsten Punktzahlen würden die Ausgangspopulation bilden (Meadows *et al.*, 2013). Die Alternative wäre, diese gesamte Population nach dem Zufallsprinzip zu erzeugen, was von der Komplexität des Problems abhängt und die Beachtung von Diversität und Auswahl erfordert (Diaz-Gomez und Hougen, 2007).

Die Bewertungsfunktion (Zielfunktion, Eignungsfunktion oder Fitnessfunktion) schätzt die Fähigkeit des Individuums, das Problem zu lösen (Biancardi und Mestria, 2024), und drückt somit notwendigerweise die Anforderungen an die Anpassung der Population aus. Mit anderen Worten, sie sollte genau zwischen geeigneten Individuen unterscheiden und die besten Lösungen bevorzugen (Eiben und Smith, 2015), was bedeuten würde, dass die Menge der Individuen gefunden wird, die die am besten angepassten Nachkommen haben, die die nächste Generation bilden. Diese Auswahl hängt von der Selektionsmethode ab, für die folgende Möglichkeiten bestehen (Linden, 2012):

Auswahl nach Fitness: Es würden die Individuen mit den höchsten Werten in der Bewertungsfunktion ausgewählt, so dass die fittesten immer die nächste Generation bilden würden. Dies würde eine vorzeitige Konvergenz begünstigen, da sich die Suche auf einen bestimmten Bereich des Raums konzentrieren würde, wodurch die globale Suche tendenziell undurchführbar würde und das Ergebnis ein lokales Optimum wäre;

Auswahl im Roulette-Rad: Je nach Wert der Bewertungsfunktion würde jedem Individuum ein proportionaler Bereich im Roulette-Rad zugewiesen. Ausgewählt werden die Individuen, bei denen das Roulette-Rad nach dem Drehen stehen bleibt, so dass sie in der

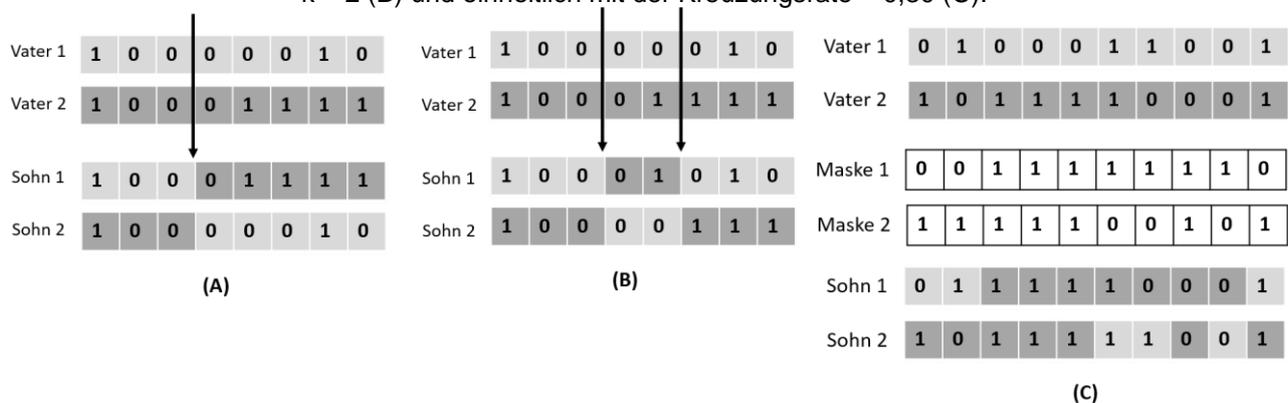
nächsten Generation zu Eltern werden. Die Anzahl der Durchläufe wäre gleich der festgelegten Anzahl der Individuen in der Population;

Auswahl nach Rangfolge: Dies würde eine vorzeitige Konvergenz vermeiden, da die Individuen nach dem Wert der Bewertungsfunktion geordnet würden und die Wahrscheinlichkeit der Auswahl durch die Reihenfolge der Individuen bestimmt würde;

Auswahl durch Turnier: die vielleicht am häufigsten verwendete Methode. N Individuen werden nach dem Zufallsprinzip ausgewählt. Das Turnier wird durch den Vergleich der Bewertungsfunktion von zwei Individuen durchgeführt, und dasjenige mit dem höchsten Wert wird ausgewählt. Es gibt auch eine Variante, bei der eine Konstante verwendet wird, zum Beispiel $Z = 0,82$, die bei jedem Turnier mit einer Zufallszahl (A) zwischen Null und Eins verglichen wird. Wenn $A < Z$ des Paares von Individuen ist, würde dasjenige mit dem höheren Bewertungswert ausgewählt werden, andernfalls wäre es ihr Gegner.

Um effektiv eine neue Generation zu erhalten, müssten die Individuen durch Crossover kombiniert werden, um die Suche zu verfeinern, und durch Mutation, um den Suchbereich zu erweitern. Diese Operatoren würden eine Erhöhung der genetischen Variabilität in der nachfolgenden Generation fördern (Dréo *et al.*, 2010). Beim Crossover werden zwei Individuen ausgewählt und Teile ihrer Gene nach dem Zufallsprinzip ausgetauscht. Dies kann auf verschiedene Weise geschehen, aber die am häufigsten verwendeten Methoden sind (Chaudhry und Usman, 2017; Umbarkar und Sheth, 2015):

Abbildung 1: Darstellung der Methoden für das Kreuzen eines einzelnen Punktes (A), mehrerer Punkte mit $k = 2$ (B) und einheitlich mit der Kreuzungsrate = 0,30 (C).



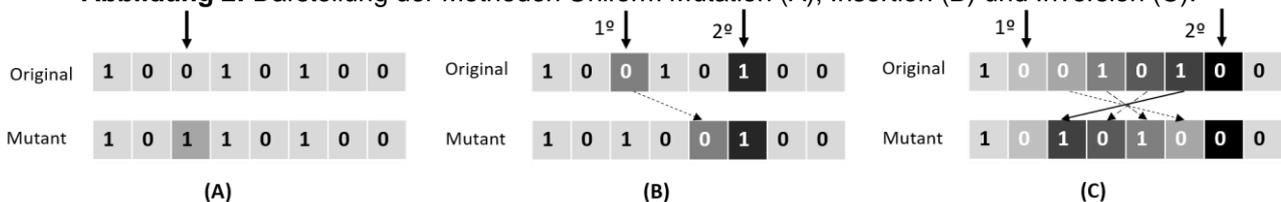
Quelle: Die Autoren (2025).

Die Mutation würde bei zufällig ausgewählten Individuen stattfinden, die als Eltern gewählt werden (Ribeiro und Brasil, 2025; Cossi und Afonso, 2024; Simões und Ebecken,

2016) und deren jeweilige Repräsentationen zufällig durch eine der folgenden Methoden verändert werden (Eiben und Smith, 2015; Soni und Kumar, 2014):

- Uniform: wird in jeder ganzzahligen oder reellen Darstellung verwendet und besteht darin, den Wert eines Gens unter Beachtung des festgelegten Bereichs zu ändern, wobei beide (Wert und Gen) zufällig ausgewählt werden (Abbildung 2A);
- Einfügung: bei der Permutation werden im Wesentlichen zwei Gene ausgewählt, wobei das erste an die Position vor dem zweiten Gen verschoben wird, während die anderen nach links verschoben werden (Abbildung 2B);
- Inversion: Bei der Permutation werden ebenfalls zwei Gene nach dem Zufallsprinzip ausgewählt, wobei die Position aller Gene untereinander getauscht wird (Abbildung 2C).

Abbildung 2: Darstellung der Methoden Uniform Mutation (A), Insertion (B) und Inversion (C).



Quelle: Die Autoren (2025).

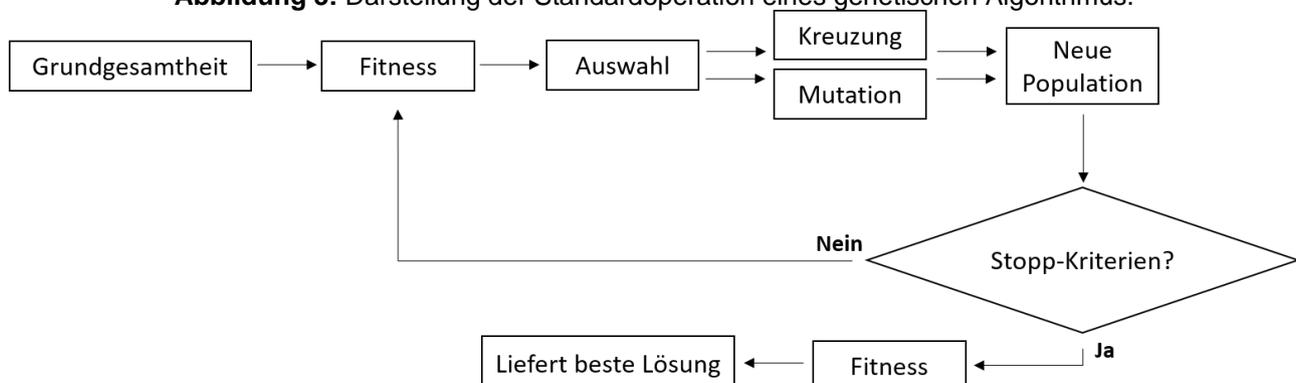
Wie bei jedem Algorithmus muss ein Stoppkriterium festgelegt werden, das das Erreichen des optimalen Wertes oder die Erfüllung einer Bedingung sein kann, wie zum Beispiel die maximale Ausführungszeit, die maximale Anzahl von Generationen, die maximale Anzahl von Anwendungen der Bewertungsfunktion oder das Nichtvorhandensein von Veränderungen durch die genetischen Operatoren (Cappellini, 2017; Michalewicz, 2013). Dann mussten zwangsläufig notwendige und operationell sensible Parameter in Übereinstimmung mit dem Kontext des untersuchten Phänomens konfiguriert werden (Linden, 2012; Goldberg, 1989):

- Anzahl der Generationen: Wenn sie klein ist, wird in der Regel keine zufriedenstellende Lösung gefunden, wenn sie jedoch hoch ist, würde sie die Rechenzeit beeinträchtigen;
- Populationsgröße: gibt die Anzahl der Individuen pro Generation an, die konstant oder variabel sein kann. Sie kann konstant oder variabel sein. Der Einfluss auf den GA ist ähnlich wie der Einfluss der Anzahl der Generationen;

- Crossover-Wahrscheinlichkeit: gibt die Wahrscheinlichkeit an, dass Individuen genetisches Material austauschen. Ist sie hoch, begünstigt sie die Entstehung neuer Merkmale, aber angepasste Individuen könnten ersetzt werden;
- Mutationswahrscheinlichkeit: gibt die Wahrscheinlichkeit an, dass Individuen verändert werden. Normalerweise wird ein niedriger Wert festgelegt, dies ist jedoch keine Regel;
- Größe des Turniers: gibt die Anzahl der für das Turnier ausgewählten Individuen an. Sie steht in direktem Zusammenhang mit dem Selektionsdruck, denn wenn sie hoch ist, begünstigt sie die am besten angepassten Individuen, wenn sie jedoch niedrig ist, erhöht sie die Zufälligkeit der Selektion und verringert den Selektionsdruck. Dies wäre die Intensität der Selektion, mit anderen Worten, die Variabilität der durchschnittlichen Fitness der Population (Blickle, 1996).

Sobald die oben genannten Kriterien festgelegt sind, wird die Standard-GA-Operation darin bestehen, dass die ursprüngliche Population der Bewertungsfunktion unterzogen wird, wodurch die besten Individuen ausgewählt werden können. Diese werden dann den Operatoren Mutation und Crossover unterworfen, um die neue Population zu erzeugen. Anschließend wird das Stopp-Kriterium überprüft, und wenn es erfüllt ist, werden die Werte der Bewertungsfunktion geschätzt und die besten Ergebnisse angezeigt. Andernfalls beginnt der gesamte Prozess wieder beim ersten Durchlauf der Anpassungsfunktion (Abbildung 3). Auf der Grundlage der obigen Ausführungen bestand das Ziel der Studie in der Entwicklung des GA-Modells zur Auswahl der besten Kombination von Sportlern.

Abbildung 3: Darstellung der Standardoperation eines genetischen Algorithmus.



Quelle: Die Autoren (2025) adaptiert von Izidoro *et al.* (2014).

Methodik

Die Gruppe der Freiwilligen bestand aus 10 männlichen Volleyballspielern im Alter zwischen 16 und 18 Jahren. Sie alle waren Kandidaten, die von einem Verein in der Stadt Rio de Janeiro eingestellt werden sollten. Die Organisation stufte die Spieler nach Kosten und potenziellem Ertrag (Punkte) ein. Die Kosten umfassten die Ausgaben und Investitionen für Transport, Verpflegung, Studium, Gesundheit und Uniformen, deren Summe mit einer direkten Beziehung auf das Intervall [1; 10] normiert wurde. Die Variable Punkte wurde ähnlich behandelt, berücksichtigte jedoch Aspekte wie technische und taktische Fähigkeiten, körperliche Kondition, potenzieller künftiger Verkaufswert, Imageverwertung und allgemeine Verhandlungsführung (Tabelle 1). Diese Parameter wurden aufgrund der strategischen Wahrnehmung (Zukunftsvision) der für die Auswahl im Zielverein Verantwortlichen berücksichtigt, was die Verwendung anderer Parameter wie Gesundheit, Ernährung, funktionelle Variablen, Verletzungsgeschichte, anthropometrische Merkmale, Alter oder Ergebnisse physischer, psychometrischer und biochemischer Tests nicht ausschließt (Freitas, 2024; Félix, 2017; Nascimento und Alencar, 2007).

Tabelle 1: Athletendaten als Eingabe für das Modell des genetischen Algorithmus.

Sportler	Kosten	Punkte
A1	5	7
A2	9	5
A3	6	3
A4	5	10
A5	8	4
A6	6	2
A7	9	8
A8	2	6
A9	4	7
A10	7	5
Gesamt	61	57

Quelle: Die Autoren (2025).

Die Zielfunktion musste die Variable Punkte unter der Kostenbeschränkung maximieren, die nicht größer als 52 sein durfte, was 85,52 % der Gesamtkosten entspricht. Die Kodierung wurde in RStudio 2022.07.1+554 „Spotted Wakerobin“ Release (7872775ebddc40635780ca1ed238934c3345c5de, 2022-07-22) für Windows entwickelt, wobei das Paket GA 3.2.4 (*Genetic Algorithms*) zur Modellierung des genetischen Algorithmus verwendet wurde. Um die Anzahl der Kombinationen zu ermitteln, die die

gestellten Bedingungen erfüllen, wurde das Paket `gtools 3.9.5` (*Various R Programming Tools*) verwendet.

Ergebnisse und Diskussion

Zunächst musste die Anzahl der möglichen Kombinationen des untersuchten Problems geschätzt werden (I), dann wurde die Kombinationsformel ohne Wiederholungen (Morgado, 1997) angewandt, wobei die Anzahl der verfügbaren Spieler, $n = 10$, und die Anzahl der auswählbaren Spieler, $k \in \{1, 2, \dots, 10\}$, berücksichtigt wurde, da es keine Beschränkung für die Anzahl der Spieler gab, die eingestellt werden konnten, was zu 1023 möglichen Lösungen führte (II).

$$\sum_{k=1}^{10} C(n, k) = \sum_{k=1}^{10} \frac{n!}{k! (n-k)!} \quad (I)$$

$$\sum_{k=1}^{10} C(10, k) = \sum_{k=1}^{10} \frac{10!}{k! (10-k)!}$$

$$\begin{aligned} \sum_{k=1}^{10} C(10, k) &= \frac{10!}{1!(10-1)!} + \frac{10!}{2!(10-2)!} + \frac{10!}{3!(10-3)!} + \frac{10!}{4!(10-4)!} + \frac{10!}{5!(10-5)!} \\ &+ \frac{10!}{6!(10-6)!} + \frac{10!}{7!(10-7)!} + \frac{10!}{8!(10-8)!} + \frac{10!}{9!(10-9)!} + \frac{10!}{10!(10-10)!} \end{aligned}$$

$$\sum_{k=1}^{10} C(10, k) = 10 + 45 + 120 + 210 + 252 + 210 + 120 + 45 + 10 + 1 = 1023 \quad (II)$$

Einige dieser Kombinationen würden jedoch die Kostenbeschränkung nicht erfüllen (Kosten ≤ 52 oder Kosten < 53), so dass beschlossen wurde, die Anzahl der Kombinationen zu schätzen, die die Beschränkung nicht verletzen würden, wobei Tabelle 1 berücksichtigt wurde. Zu diesem Zweck wurde das folgende Programm entwickelt. In den ersten drei Zeilen haben wir den Vektor mit dem Wert der Kosten jedes Athleten (*elementos*) und die Variablen `Limit` für die Summe und den Zähler erstellt, wobei letzterer die Anzahl der Kombinationen mit Kosten < 53 akkumuliert. Wir haben die Funktion

gerador_combinacoes (Zeilen 4 bis 20) codiert, um die 1023 möglichen Kombinationen zu generieren und sie einzeln zu testen. Jedes Mal, wenn eine Kombination die Restriktion erfüllte, d.h. die Bedingung in Zeile 15 wahr war, wurde die Zählervariable um eine Einheit erhöht. Am Ende würde das Programm die gesuchte Menge anzeigen, nämlich 1009 Kombinationen. Diese Menge war notwendig und ausreichend, um die Entwicklung des GA-Modells zur Ermittlung des besten Ergebnisses für das vorgeschlagene Problem zu rechtfertigen.

```

1  elementos <- c(5, 9, 6, 5, 8, 6, 9, 2, 4, 7)
2  limite_soma <- 53
3  contador <- 0
4  gerador_combinacoes <- function(elementos, limite_soma) {
5    n <- length(elementos)
6    for (i in 1:(2^n - 1)) {
7      combinacoes <- integer(0)
8      soma_combinacoes <- 0
9      for (j in 0:(n-1)) {
10       if (bitwAnd(i, bitwShiftL(1, j))) {
11         combinacoes <- c(combinacoes, elementos[j+1])
12         soma_combinacoes <- soma_combinacoes + elementos[j+1]
13       }
14     }
15     if (soma_combinacoes < limite_soma) {
16       contador <<- contador + 1
17     }
18   }
19   return(contador)
20 }
21 resultado <- gerador_combinacoes(elementos, limite_soma)
22 print(paste("Número de combinações com soma menor que 53:", resultado))

```

Das GA-Modell wurde gemäß der folgenden Kodierung entwickelt, und die Zeilen wurden mit Erläuterungen (Kommentaren) versehen, die mit dem Symbol # beginnen. Es ist erwähnenswert, dass die Namen und Eigenschaften der Athleten durch die Vektoren *Athlet*, *Punkt* und *Kosten* dargestellt wurden, die den Datenrahmen (Tabellenkalkulationsstruktur) namens *moneyball* bildeten. Die Auswertungsfunktion (*funcao_avaliacao*) prüfte die Einhaltung der Beschränkung und wählte die Sportler aus, wobei sie den Wert null zurückgab, wenn die Auswahl nicht erfolgte, und andernfalls 1.

Auch in Bezug auf den Code hatte die Verwendung der *ga*-Funktion aus dem GA-Paket das Ergebnis in der *resultado_ga*-Variable akkumuliert. Die Parameter des

entwickelten Modells wurden auf der Grundlage des untersuchten Phänomens und der Erfahrung des Programmierers festgelegt. Der zugewiesene Typ war binär (*binary*), da das Ziel darin bestand, Athleten auszuwählen, die vorkommen können oder nicht. Die Anzahl der Bits (*nBits*) würde immer der Anzahl der Spieler (Elemente oder Items) entsprechen. Die Ausgangspopulation (*popSize*) sollte aus 50 Individuen bestehen, und das Modell sollte in bis zu 100 Generationen konvergieren (*maxiter*), von denen jeweils nur die beiden Individuen mit den höchsten Werten in der Bewertungsfunktion in die nächste Generation übergehen würden (Elitismus, *elitism*). Den Operatoren *pmutation* und *pcrossover* wurden Wahrscheinlichkeiten von 0,10 bzw. 0,80 zugewiesen. Der Algorithmus begann mit dem Keim (*seed*), der die Reproduzierbarkeit der Ergebnisse gewährleisten sollte.

```

1 # Laden Sie das GA-Paket.
  library(GA)
2 # Definieren Sie die Daten
  atleta <- c("a1", "a2", "a3", "a4", "a5", "a6", "a7", "a8", "a9", "a10")
  ponto <- c(7, 5, 3, 10, 4, 2, 8, 6, 7, 5)
  custo <- c(5, 9, 6, 5, 8, 6, 9, 2, 4, 7)
3 # Erstellen Sie den Datenrahmen
  moneyball <- data.frame(atleta, ponto, custo)
4 # Definieren Sie die Auswertungsfunktion.
  # Erhält einen binären Vektor (0 oder 1), der angibt, ob der Athlet ausgewählt wurde
  # oder nicht.
  # Gibt die Summe der Punkte zurück, wenn die Gesamtkosten <= 52 sind,
  # andernfalls wird 0 zurückgegeben.
  funcao_avalicao <- function(x) {
    custo_total <- sum(moneyball$custo * x) # Berechnet die Gesamtkosten für die
    ausgewählten Sportler
    if (custo_total > 53) {
      return(0) # Gibt 0 zurück, wenn die Kosten den Grenzwert überschreiten
    } else {
      return(sum(moneyball$ponto * x)) # Gibt die Summe der Punkte zurück
    }
  }
5 # Definieren Sie die Parameter des genetischen Algorithmus
  resultado_ga <- ga(type = "binary", nBits = nrow(moneyball),
    fitness = funcao_avalicao, popSize = 50, maxiter = 100,
    pmutation = 0.1, pcrossover = 0.8, elitism = 2, seed = 123)
6 # Ausgewählte Athleten anzeigen
  atletas_selecionados <- moneyball$atleta[resultado_ga@solution == 1]
  cat("Atletas selecionados:", paste(atletas_selecionados, collapse = ", "), "\n")
7 # Den gefundenen Maximalwert anzeigen
  cat("Valor máximo encontrado:", max(resultado_ga@summary[, "max"]), "\n")
8 # Zeigen Sie das Diagramm „Auswertungs- vs. Generierungsfunktion“
  plot(resultado_ga)

```

Das GA-Modell (Tabelle 2) zeigte die Auswahl von acht Athleten an, die nicht aus den Gruppen A3 und A6 stammten, so dass die erreichten Kosten 49 (= 5 + 9 + 5 + 8 + 9 + 2 + 4 + 7) betragen, was 94,23 % der Beschränkung (Kosten < 53) und 80,33 % der theoretischen Kosten (61) entsprach, was beweist, dass die auferlegten Bedingungen nicht verletzt wurden. Die Auswahlmenge ergab Punkte = 52 (= 7 + 5 + 10 + 4 + 8 + 6 + 7 + 5), was etwa 91,23 % der Höchstpunktzahl entspricht. Somit hat das vorgeschlagene Modell angemessen auf die Anfrage reagiert und die Entscheidungsfindung wirksam unterstützt.

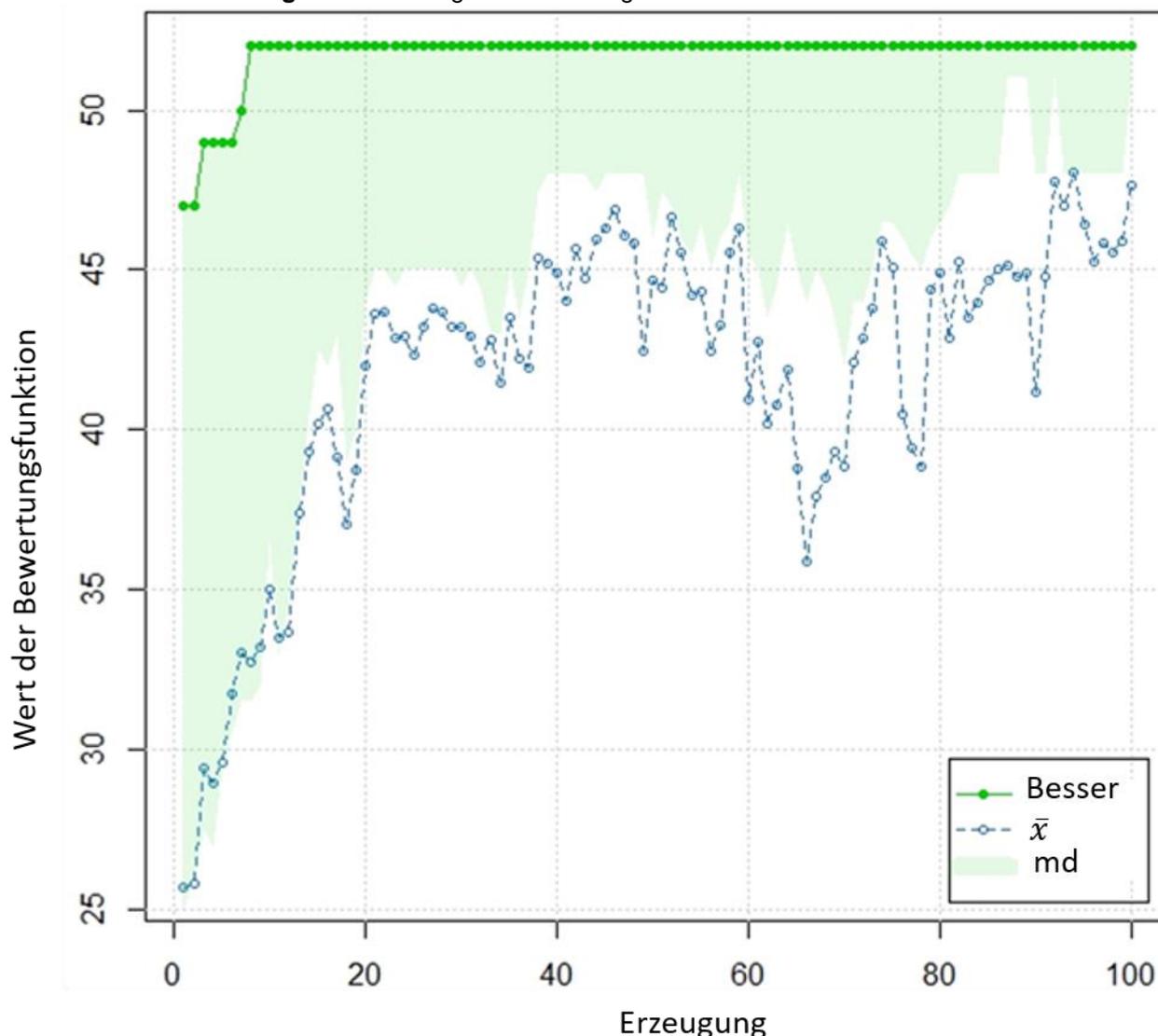
Tabelle 2: Ergebnisse der Athletenauswahl mit dem Modell des Genetischen Algorithmus.

Sportler	Kosten	Punkte	Ausgewählt
A1	5	7	1
A2	9	5	1
A3	6	3	0
A4	5	10	1
A5	8	4	1
A6	6	2	0
A7	9	8	1
A8	2	6	1
A9	4	7	1
A10	7	5	1
Gesamt	61	57	8

Quelle: Die Autoren (2025).

Es ist erwähnenswert, dass in allen Generationen das bestmögliche Ergebnis immer höher war als das Durchschnittsergebnis (Abbildung 4), was eine hohe Variabilität zeigt. Diese Eigenschaft war auch beim Median vorhanden, aber dieser lag erwartungsgemäß näher am besten Ergebnis, als Folge des fehlenden Einflusses von Extremwerten (Triola, 2017). Diese Feststellung wurde durch das Erreichen des besten Ergebnisses in der achten Generation bestätigt, was möglicherweise auf die geringe Komplexität des untersuchten Phänomens zurückzuführen ist. Dies führte dazu, dass das Stoppkriterium, Anzahl der Generationen = 100, zu hoch angesetzt wurde, und die Verringerung der Variabilität über die Generationen hinweg deutete darauf hin, dass die durch die genetischen Operatoren hervorgerufenen Veränderungen tendenziell unbedeutend waren (Cappellini, 2017; Michalewicz, 2013).

Abbildung 4: Entwicklung der Bewertungsfunktion über die Generationen.



Quelle: Die Autoren (2025).

Abschließende Überlegungen

Ziel dieser Arbeit war es, ein Modell eines genetischen Algorithmus (GA) zu entwickeln, um aus 10 Kandidaten die größte Anzahl von Volleyballspielern auszuwählen. Die Ergebnisse zeigten, dass trotz der geringen Komplexität des Phänomens die Anzahl der Kombinationen (1009 Möglichkeiten) für Versuch und Irrtum zu hoch war. Das Modell konvergierte jedoch bereits in der achten Generation zum optimalen Ergebnis, was einen geringen Rechenaufwand beweist. Daraus lässt sich schließen, dass das Ziel zufriedenstellend erreicht wurde.

Für zukünftige Studien empfehlen wir, die Komplexität des Modells zu erhöhen und die Auswahl der Spieler unter Einbeziehung der Spielposition festzulegen, was die

Bewertungsfunktion verändern würde. Die Indizes Kosten und Punkte könnten ohne Normalisierung verwendet werden, wobei die realen Werte genommen werden, was zu anderen Ergebnissen führen könnte. Änderungen der Wahrscheinlichkeiten der genetischen Operatoren und des Elitismus sowie die Auswahl durch ein Turnier könnten eine Bereicherung der methodologischen Diskussion über genetische Algorithmen darstellen.

Referenzen

BIANCARDI, GS; MESTRIA, M. Algoritmo genético para localização de agregadores de dados em redes elétricas inteligentes. **VETOR - Revista de Ciências Exatas e Engenharias**, v. 34, n. 1, p. 130-144, 2024.

BLICKLE, T. **Theory of evolutionary algorithms and application to system synthesis**. Ph.D. dissertation (Doctor of Technical Sciences) - Swiss Federal Institute of Technology, ETH. Zurich (Switzerland), 1996.

CAPPELLINI, L. **Algoritmi per la determinazione di posizione ed intensità di un impatto su strutture composite**. Laurea Magistrale in Ingegneria Aerospaziale. Dipartimento di Ingegneria Industriale. Università Degli Studi Di Padova. Padova (Italian), 2017.

CHAUDHRY, IA; USMAN, M. Integrated process planning and scheduling using genetic algorithms. **Tehnički Vjesnik**, v. 24, n. 5, p. 1401-1409, 2017.

COSSI, AM; AFONSO, GF. Algoritmo BVNS aplicado no problema de restauração de redes de distribuição de energia elétrica, considerando programa de resposta da demanda. **Trends in Computational and Applied Mathematics**, v. 25, e01690, 2024.

DIAZ-GOMEZ, PA; HOUGEN, DF. Initial population for genetic algorithms: A metric approach. In **Proceedings of the 2007 International Conference on Genetic and Evolutionary Methods**. Las Vegas (USA): CSREA Press, 2007, p. 43–49.

DRÉO, J *et al.* **Metaheuristics for hard optimization: methods and case studies**. Berlin (Deutschland): Springer, 2010.

EIBEN, AE; SMITH, JE. **Introduction to evolutionary computing**. Berlin (Deutschland): Springer, 2015.

FÉLIX, GS. **Relação de testes psicométricos com variáveis fisiológicas utilizadas no controle das cargas de treino em atletas recreacionais**. Dissertação (Mestrado em Educação Física) – Programa Associado de Pós-graduação em Educação Física. Universidade Federal da Paraíba. João Pessoa (PB), 2017.

FREITAS, HB. **Efeitos do Pré-Condicionamento Isquêmico (PCI) sobre composição corporal, variáveis neuromusculares e bioquímicas em atletas paralímpicos em período de destreinamento**. Dissertação (Mestrado em Fisioterapia) – Programa de Pós-graduação em Fisioterapia. Universidade Federal da Paraíba. João Pessoa (PB), 2024.

GOLDBERG, D. **Genetic algorithms in search, optimization, and machine learning**. New York (USA): Addison-Wesley Publishing Company, 1989.

IZIDORO, SC *et al.* GASS: identifying enzyme active sites with genetic algorithms. **Bioinformatics**, v. 31, n. 6, p. 864–870, 2014.

KATOCH, S; CHAUHAN, SS; KUMAR, V. A review on genetic algorithm: past, present, and future. **Multimedia Tools and Applications**, v. 80, n. 5, p. 8091-8126, 2021.

L'ECUYER, P. Uniform random number generation. **Annals Operations Research**, v. 53, n. 1, p. 77-120, 1994.

LINDEN, R. **Algoritmos genéticos**. Rio de Janeiro: Ciência Moderna, 2012.

MEADOWS, B *et al.* Evaluating the seeding genetic algorithm. In CRANFIELD, S; NAYAK, A. (Eds.). **AI 2013: Advances in Artificial Intelligence**. Berlin (Deutschland): Springer, 2013, p. 221–227.

MICHALEWICZ, Z. Genetic algorithms + data structures = evolution programs. Berlin (Deutschland): Springer, 2013.

MICHALEWICZ, Z; FOGEL, DB. **How to solve it: modern heuristics**. Berlin (Deutschland): Springer, 2013.

MORGADO, ACO. **Análise combinatória e probabilidade: com as soluções dos exercícios**. Rio de Janeiro: Sociedade Brasileira de Matemática, 1997.

NASCIMENTO, OV; ALENCAR, FH. Perfil do estado nutricional do atleta adulto. **Fitness & Performance Journal**, v. 6, n. 4, p. 241-246, 2007.

PARK, SK; MILLER, KW. Random number generators: good ones are hard to find. **Communications of the ACM**, v. 31 n. 10, p. 1192-1201, 1988.

PINTO, ARF; MARTARELLI, NJ; NAGANO, MS. Algoritmo genético: principais *gaps*, *trade-offs* e perspectivas para futuras pesquisas. **Trends in Computational and Applied Mathematics**, v. 23, n. 3, p. 413-438, 2022.

RIBEIRO, BB; BRASIL, CRS. Um estudo comparativo de algoritmos genéticos aplicados ao problema das N rainhas. **Proceeding Series of the Brazilian Society of Computational and Applied Mathematics**, v. 11, n. 1, p. 1-2, 2025.

RUSSELL, S; NORVIG, P. **Inteligência artificial: uma abordagem moderna**. Rio de Janeiro: GEN LTC, 2022.

SÁEZ-GUTIÉRREZ, FL *et al.* Reconstrucción de la superficie corneal del ojo humano mediante un algoritmo evolutivo computacional. Aplicación práctica en casos no patológicos. **DYNA**, v. 99, n. 1, p. 85-92, 2024.

SIMÕES, GJ; EBECKEN, NFF. Algoritmo genético e enxame de partículas para a otimização de suportes laterais de fornos. **Revista Internacional de Métodos Numéricos para Cálculo y Diseño en Ingeniería**, v. 32, n. 1, p. 7-12, 2016.

SONI, N; KUMAR, T. Study of various mutation operators in genetic algorithms. **(IJCSIT) International Journal of Computer Science and Information Technologies**, v. 5, n. 3, p. 4519–4521, 2014.

UMBARKAR, AJ; SHETH, PDS. Crossover operators in genetic algorithms: a review. **ICTACT Journal on Soft Computing**, v. 6, n. 1, p. 1083–1092, 2015.