
ДИАЛОГИ СО СПЕЦИАЛИСТАМИ

Научная статья

УДК 681.518; DOI: 10.61260/2304-0130-2026-1-46-51

ГЕНЕТИЧЕСКИЕ АЛГОРИТМЫ И ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ

✉ **Лабинский Александр Юрьевич.**

Санкт-Петербургский университет ГПС МЧС России, Санкт-Петербург, Россия

✉ labinskyi.a@igps.ru

Аннотация. Рассмотрены возможности использования генетических алгоритмов (ГА) в качестве основы для создания систем искусственного интеллекта (ИИ). Преимуществами ГА является универсальность, то есть возможность использования для решения различных задач, где требуется нахождение оптимального решения задачи, а также способность к работе с неполными или зашумленными данными.

Подробно рассмотрены особенности поиска оптимального решения, основанного на эволюционном процессе, и алгоритм работы ГА, который содержит основные этапы работы ГА: создание начальной популяции, отбор наиболее приспособленных особей по значению функции приспособленности (целевой функции), скрещивание (рекомбинация) и мутация.

Рассмотрено масштабирование функции приспособленности (целевой функции), выполняемое одним из трех методов: линейное масштабирование, степенное масштабирование и сигма-отсечение. Подробно рассмотрены методы кодирования информации в ГА, включая целочисленное, вещественное и логарифмическое кодирование. Особое внимание уделено вопросу сходимости ГА, зависящей от правильного, на начальном и конечном этапах работы ГА, использования на шаге рекомбинации операторов мейоза и митоза. Рассмотрены области применения ГА, а также ограничения, присущие генетическим алгоритмам.

Ключевые слова: искусственный интеллект, генетический алгоритм, кодирование информации, масштабирование, мутация, популяция, рекомбинация, отбор, функция приспособленности, целевая функция, оптимизация

Для цитирования: Лабинский А. Ю. Генетические алгоритмы и искусственный интеллект // Надзорная деятельность и судебная экспертиза в системе безопасности. 2025. № 1. С. 46–51. DOI: 10.61260/2304-0130-2026-1-46-51

Введение

Одним из подходов к созданию искусственного интеллекта (ИИ) является подход, связанный с созданием моделей ИИ, основанных на генетических алгоритмах (ГА), где адаптация к решаемой задаче производится путем создания популяций соревнующихся между собой решений, которые совершенствуются на основе природного механизма эволюции.

ГА используют концепции из эволюционной биологии для поиска оптимальных решений. Поэтому ГА являются мощным инструментом оптимизации и находят применение во многих областях, начиная от науки и техники и заканчивая экономикой и искусством.

Сформулируем постановку задачи для данной статьи. Нужно рассмотреть подход к решению задачи создания искусственного интеллекта, основанный на использовании компьютерной симуляции эволюционных процессов.

Тема статьи актуальна, так как проблема разработки систем ИИ, которые могут использоваться, в том числе, для оценки риска и принятия решений, относится к актуальным проблемам. Актуальность проблемы разработки систем ИИ постоянно увеличивается и на тему ИИ написано много работ, в том числе работы [1, 2].

Понятие генетического алгоритма

Поиск оптимального решения, основанный на эволюционном процессе, реализующем механизм наследственности, был предложен Холландом в 1975 году и послужил основой для создания генетического алгоритма. ГА использует метаэвристический метод оптимизации, состоящий из процессов естественного отбора и генетической мутации. Основная идея состоит в создании популяции индивидуальных решений (часто называемых хромосомами), их оценке на соответствие заданным критериям и последующем изменении путем скрещивания и мутации. Эти процессы схожи с тем, как в природе эволюционируют живые организмы [3–6].

Процесс работы ГА включает в себя следующие шаги [7]:

Инициализация. Начальная популяция индивидуумов создается случайным образом, выбирается некоторый критерий останова (число поколений или степень приспособленности) и запускается итерационный процесс.

Оценка. Каждый индивидуум (элемент данных) оценивается по заданному критерию (функции приспособленности), которая показывает, насколько хорошо данное решение решает задачу.

Отбор. Индивидуумы (элементы данных) отбираются на основе их приспособленности. Элементы данных с лучшей оценкой могут остаться в итерационном процессе. С целью выполнения отбора каждому элементу данных назначается вероятность P_i , равная относительной приспособленности. После этого производится отбор элементов данных с замещением данных в соответствии уменьшением величины вероятности P_i .

Скрещивание (рекомбинация). Пары индивидуумов (элементы данных) скрещиваются (рекомбинируются), создавая потомство, которое наследует части родительских признаков.

Мутация. Производятся случайные изменения признаков потомства (поколения), имитирующие мутации в геноме.

Замена. Новое поколение заменяет старое, и процесс повторяется.

Остановка. Алгоритм выполняется до тех пор, пока не будет достигнуто условие критерия останова итерационного процесса (заданное количество поколений, достижение определенной приспособленности или заданное значение целевой функции).

На рис. 1 представлена схема работы типового генетического алгоритма:

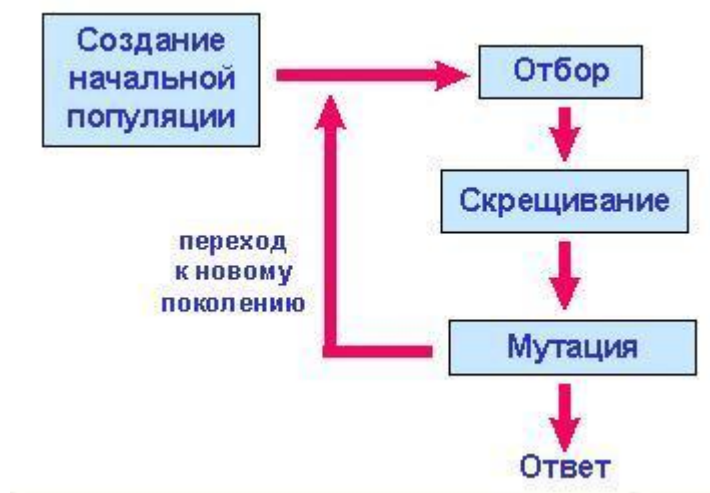


Рис. 1. Схема работы ГА

Поиск решения задачи в генетическом алгоритме

Генетические алгоритмы реализуют процедуру поиска решения задачи (часто задачи оптимизации), основанную на механизмах естественного отбора и наследования и использующую принцип выживания наиболее приспособленных особей. В процессе поиска ГА одновременно использует два метода: метод отбора особей с лучшей приспособленностью, используя генетический оператор размножения (кроссовер), и метод мутации особей.

Указанная процедура поиска генетического алгоритма (ППГА) имеет несколько отличий от традиционных методов поиска решений [6]:

– ППГА обрабатывает закодированную форму параметров, а не сами значения параметров самой задачи, в соответствии со свойством ГА – необходимостью кодирования параметров;

– ППГА осуществляет поиск решения исходя из некоторого множества (популяции), а не из единственной точки, в соответствии со свойством ГА – необходимостью операций на популяциях;

– ППГА использует целевую функцию (функцию приспособленности);

– ППГА применяет не детерминированные, а вероятностные правила отбора, в соответствии со свойством ГА – рандомизацией операций.

Данные свойства ГА по сравнению с другими методами поиска решений обеспечивают некоторые преимущества ГА, так как приводят в результате к устойчивости ГА.

Кодирование информации в генетических алгоритмах

Представление параметров решаемой задачи в хромосомном виде влияет на выбор исходной популяции генетического алгоритма. Хромосомный вид параметров предполагает двоичное представление всех генов в хромосоме (аллели генов равны 0 или 1). Количество расчетных точек в пространстве поиска влияет на длину хромосомы.

Для целочисленной переменной X , принимающей 32 значения в интервале $[1, 32]$, рассмотрим пример нахождения максимума функции $f(x) = 2 \cdot x^2 + 3$. В соответствии с представлением чисел в двоичной системе счисления целые числа из этого интервала можно представить в виде последовательностей нулей и единиц. В этом случае длина хромосомы будет составлять 5 бит ($2^5 = 32$).

Для определения последовательности элементов бинарного множества, состоящего из N элементов, используем алгоритм построения бинарного кода Грея. В данном алгоритме каждое следующее подмножество получается из предыдущего путем удаления или добавления одного элемента. Алгоритм построения бинарного кода Грея задает последовательность всех подмножеств (размером 5 бит вида 10101) N -элементного множества, где $N = 32$ подмножества.

На рис. 2 представлено построение бинарного кода по алгоритму Грея с помощью консольной программы для ЭВМ.

Будем использовать целевую функцию $f(x) = 2 \cdot x^2 + 3$ в роли функции приспособленности хромосомы. В качестве исходной популяции возьмем случайным образом выбранную популяцию, состоящую из 6 кодовых последовательностей (хромосом): CHR1[10101]; CHR2[01000]; CHR3[11101]; CHR4[10011]; CHR5[00011]; CHR6[00111].

Фенотипы указанных исходных хромосом равны десятичным числам, соответствующим двоичным последовательностям:

$$\text{CHR1}^* = 1 \cdot 2^0 + 0 \cdot 2^1 + 1 \cdot 2^2 + 0 \cdot 2^3 + 1 \cdot 2^4 = 21; \text{CHR2}^* = 8; \text{CHR3}^* = 29;$$

$$\text{CHR4}^* = 19; \text{CHR5}^* = 3; \text{CHR6}^* = 7.$$

```

РАСЧЕТ БИНАРНОГО КОДА
по алгоритму Грея.
Число комбинаций равно 2^N.
Число разрядов N от 1 до 8.
Введите число разрядов N: 3
Число комбинаций равно 2^3 = 8
N P A[i]
1: 0; 0; 0; 0;
2: 1; 0; 0; 1;
3: 2; 0; 1; 1;
4: 1; 0; 1; 0;
5: 3; 1; 1; 0;
6: 1; 1; 1; 1;
7: 2; 1; 0; 1;
8: 1; 1; 0; 0;

```

Рис. 2. Бинарный код при N=3

Тогда значения функции приспособленности будут равны:

$$F(\text{CHR1}) = f(21) = 885; F(\text{CHR2}) = 131; F(\text{CHR3}) = 1685; F(\text{CHR4}) = 725;$$

$$F(\text{CHR5}) = 21; F(\text{CHR6}) = 101.$$

В ГА используются следующие типы кодирования: вещественное, целочисленное и логарифмическое.

1. Целочисленное кодирование.

$$R = g \cdot (X_{\max} - X_{\min}) / (2^m - 1) + X_{\min}; \quad g = (R - X_{\min}) \cdot (2^m - 1) / (X_{\max} - X_{\min}),$$

где R – целочисленное декодированное значение, g – целочисленное кодированное значение.

$$\text{Пример: если } g = 43981_{10}, \text{ то } R = 43981 \cdot (2-1) / (2^{16}-1) + 1 = 0,6711 + 1 = 1,6711.$$

2. Вещественное кодирование.

$$\text{Пример: если } R = 1,3275, \text{ то } g = (1,3275 - 1) \cdot (2^{16} - 1) / (2 - 1) = 0,3275 \cdot 65535 = 21462,7125 \approx 21462_{10} = 0101\ 0011\ 1101\ 0110_2.$$

Здесь R – вещественное декодированное значение,

g – вещественное кодированное значение, длина хромосомы 16 бит.

Для уменьшения длины хромосомы может применяться логарифмическое кодирование.

3. Логарифмическое кодирование.

При логарифмическом кодировании первый бит двоичной последовательности является битом знака степени показательной функции, второй бит – бит знака этой функции, а остальные биты – значение самой степени: $[AB \text{ bin}] = (-1)^B \cdot \exp^C$, где $C = (-1)^A [\text{bin}]_{10}$, $[\text{bin}]_{10}$ – десятичное значение числа, закодированного в виде двоичной последовательности bin .

Пример: пусть бинарная последовательность равна $[10110]$, тогда кодовая последовательность числа $X = (-1)^0 \cdot \exp^C$.

$$\text{Значение показателя степени экспоненты } C = (-1)^1 \cdot [110]_{10} = -1 \cdot 6 = -6.$$

$$\text{Тогда } X = 1 \cdot \exp^{-6} = 0,002478.$$

Если бинарный код равен $[01010]$, то кодовая последовательность числа X равна: $X = (-1)^1 \cdot \exp^C = -1 \cdot \exp^2 = -7,389056$.

Таким образом, с помощью 5 бит можно закодировать числа из интервала $[-e^7, e^7]$ или $[-1096, 1096]$.

Масштабирование функции приспособленности

В задачах теории игр функция приспособленности (ФП) имеет вид стоимостной функции (СФ), в задачах управления ФП имеет вид функции погрешности (ФП), а в задачах оптимизации, решаемых с помощью ГА, ФП обычно максимизируется и называется целевой функцией (ЦФ) [3].

При решении задач оптимизации на каждой итерации ГА с помощью ФП оценивается приспособленность каждой особи заданной популяции и затем создается следующая популяция особей (хромосом), элементы которой рассматриваются как элементы множества потенциальных решений задачи.

На начальном этапе итерационного процесса несколько особей могут иметь, по сравнению с остальными особями популяции, значительно большие значения ФП, что характерно для малых популяций. Такое развитие событий нежелательно, так как при этом происходит преждевременная сходимость ГА к неоптимальному решению. В этом случае можно изменить метод скрещивания либо использовать масштабирование ФП.

Масштабирование ФП это преобразование ФП, которое может осуществляться одним из трех методов: масштабирование с помощью сигма-отсечения, степенное масштабирование и линейное масштабирование.

Линейное масштабирование: подбираются константы a и b , входящие в уравнение $F(X)^* = a \cdot F(X) + b$, причем константы a и b подбираются так, чтобы максимальное значение ФП после масштабирования было кратным её среднему значению (коэффициент кратности от 1,2 до 2,0), а среднее значение ФП после масштабирования равнялось среднему значению ФП до масштабирования.

Степенное масштабирование: для масштабирования используется уравнение $F(X)^* = F(X)^k$, где k – число, близкое к 1 (например, 1,005).

Сигма-отсечение: для масштабирования используется уравнение $F(X)^* = F(X) + (F(X)_{\text{cp}} - c \cdot \sigma)$, где σ – стандартное отклонение по популяции, c – натуральное число от 1 до 5, $F(X)_{\text{cp}}$ – среднее значение ФП по всей популяции.

Вывод

Генетические алгоритмы могут служить в качестве основы для создания систем искусственного интеллекта. Преимуществом генетических алгоритмов является универсальность, то есть возможность использования для решения различных задач, где требуется нахождение оптимального решения задачи, а также способность к работе с неполными или зашумленными данными.

Создание автором компьютерных моделей генетических алгоритмов, представленных в работах [8–10], может рассматриваться как научная новизна исследования, отражающая личный вклад автора.

Список источников

1. Аверкин А.Н., Гаазе-Рапопорт М.Г., Поспелов Д.А. Толковый словарь по искусственному интеллекту. М.: Радио и связь, 1992. 256 с.
2. Девятков В. В. Системы искусственного интеллекта. М.: Изд-во МГТУ им. Н. Э. Баумана, 2001.
3. Гладков Л. А., Курейчик В. В, Курейчик В. М. и др. Биоинспирированные методы в оптимизации: монография. М: Физматлит, 2009.
4. Baluja S. Genetic algorithms and search statistics. MIT Pres
5. Davis L. Handbook of Genetic Algorithms. “Van Nostrand Reinhold”, 2009.
6. Michalewicz Z. Genetic Algorithms. Springer-Verlag, 2012.
7. Лабинский А.Ю., Щербаков О.В. Особенности использования компьютерной симуляции эволюционных процессов // Проблемы управления рисками в техносфере. 2017.

8. Лабинский А.Ю. Использование генетического алгоритма для многокритериальной оптимизации // Проблемы управления рисками в техносфере. № 4, 2018.
9. Лабинский А.Ю. Многопараметрическая оптимизация с помощью генетического алгоритма // Проблемы управления рисками в техносфере. № 2, 2020.
10. Лабинский А.Ю. Перспективные направления компьютерного моделирования сложных процессов и систем: монография. СПб.: СПбУ ГПС МЧС России, 2017.

Информация о статье: статья поступила в редакцию: 02.02.2026; принята к публикации: 15.02.2026

Информация об авторах:

Лабинский Александр Юрьевич, доцент кафедры прикладной математики и информационных технологий Санкт-Петербургского университета ГПС МЧС России (196105, Санкт-Петербург, Московский пр., д. 149), кандидат технических наук, доцент, e-mail: labinskyi.a@igps.ru, <https://orcid.org/0000-0001-2735-4189>, SPIN-код: 8338-4230