

Проектирование интеллектуальных систем

Лекция № 2. Генетические алгоритмы

Содержание

| | |
|--|----|
| Содержание | 1 |
| Эволюционное моделирование | 2 |
| Основные понятия генетических алгоритмов | 3 |
| Формулировка задачи в терминах ГА | 3 |
| Классический генетический алгоритм | 5 |
| Методы селекции | 6 |
| Операторы скрещивания | 9 |
| Операторы мутации | 13 |
| Пример решения задачи с помощью ГА | 13 |
| Теорема схем | 16 |
| Эвристики | 17 |
| Решение задачи коммивояжёра с помощью ГА | 19 |
| Особенности генетических алгоритмов | 23 |
| Нейроэволюция | 23 |
| Вопросы для самопроверки | 28 |
| Список литературы | 29 |

Эволюционное моделирование

Идея решать сложные задачи путём эволюции возникла в результате наблюдения за природой. Впервые идею генетических алгоритмов (ГА) высказал Дж. Холланд в 60-х годах XX века. Он предположил, что возможно реализовать компьютерную программу, которая будет решать сложные задачи путём моделирования процесса естественной эволюции. Под «сложными» задачами здесь имеются ввиду задачи с высокой вычислительной сложностью (например, задачи комбинаторной оптимизации) и плохоформализуемые задачи, которые можно свести к решению задачи оптимизации.

Общая идея решения задач при помощи генетических алгоритмов состоит в следующем.

- Формируется некоторая *популяция* начальных решений задачи (особей).
- Каждое решение представляется в виде некоторой закодированной структуры – *хромосомы*.
- Запускается моделирование процесса эволюции, которое состоит в последовательном применении операторов *селекции*, *скрещивания* и *мутации*, в результате чего формируются новые поколения решений.
- Эволюционный процесс останавливается, как только было получено удовлетворительное решение.

Основные понятия генетических алгоритмов

Многие термины, используемые в генетических алгоритмах, были позаимствованы из генетики.

Генетические алгоритмы оперируют **популяциями**, которые представляют собой множества возможных *решений* задачи.

Каждое решение – элемент популяции – называется **особью**.

Каждая особь описывается одной или несколькими **хромосомами**.

Набор хромосом каждой особи называется её **генотипом**.

Хромосома – это упорядоченная последовательность **генов**.

Ген – это атомарный элемент генотипа (единица наследственной информации).

Фенотип – это набор значений, соответствующих данному генотипу, то есть декодированные значения параметров задачи (решения).

Аллель – это значение конкретного гена (вариант свойства).

Локус – это позиция гена в хромосоме.

Для реализации естественного отбора необходим механизм оценки приспособленности каждой особи к текущим условиям окружающей среды. Для этого вводится **функция приспособленности** (англ. fitness function), которая позволяет сопоставить каждой особи некоторое число, являющееся количественной оценкой её «качества». Таким образом, решение любой задачи с помощью генетического алгоритма сводится к решению задачи оптимизации, в которой целевой функцией является функция приспособленности.

Формулировка задачи в терминах ГА

Предположим, что необходимо найти максимум функции

$$f(x) = \log(x + 5)$$

на отрезке $[0,15]$. Также предположим, что x может принимать только целочисленные значения. В таком случае множеством возможных решений задачи будет множество целых чисел от 0 до 15. Эти числа можно закодировать четырёхбитными строками. Множество возможных решений также называют **пространством поиска**.

| | | | |
|------|------|------|------|
| 0000 | 0001 | 0010 | 0011 |
| 0100 | 0101 | 0110 | 0111 |
| 1000 | 1001 | 1010 | 1011 |
| 1100 | 1101 | 1110 | 1111 |

Множество строк, представленных выше, – это популяция. Каждая строка – это особь или хромосома (эти понятия часто используются как синонимы, если особь содержит лишь одну хромосому). В данном случае хромосома одновременно задаёт генотип. Фенотип – это декодированное значение x . Например, для хромосомы 1001 фенотип – это 9.

Функция приспособленности в данном случае определяется тривиально – это будет сама оптимизируемая функция $f(x) = \log(x + 5)$. Таким образом, решение задачи сводится к выбору наиболее приспособленной особи из популяции. Очевидно, что это особь $1111_2 = 15_{10}$, так как

$$15 = \underset{0 \leq x \leq 15}{\operatorname{argmax}} \log(x + 5).$$

В данном случае не потребовалось формирование нового поколения, так как размерность пространства поиска мала для данной задачи и всё множество возможных решений вошло в начальную популяцию.

Классический генетический алгоритм

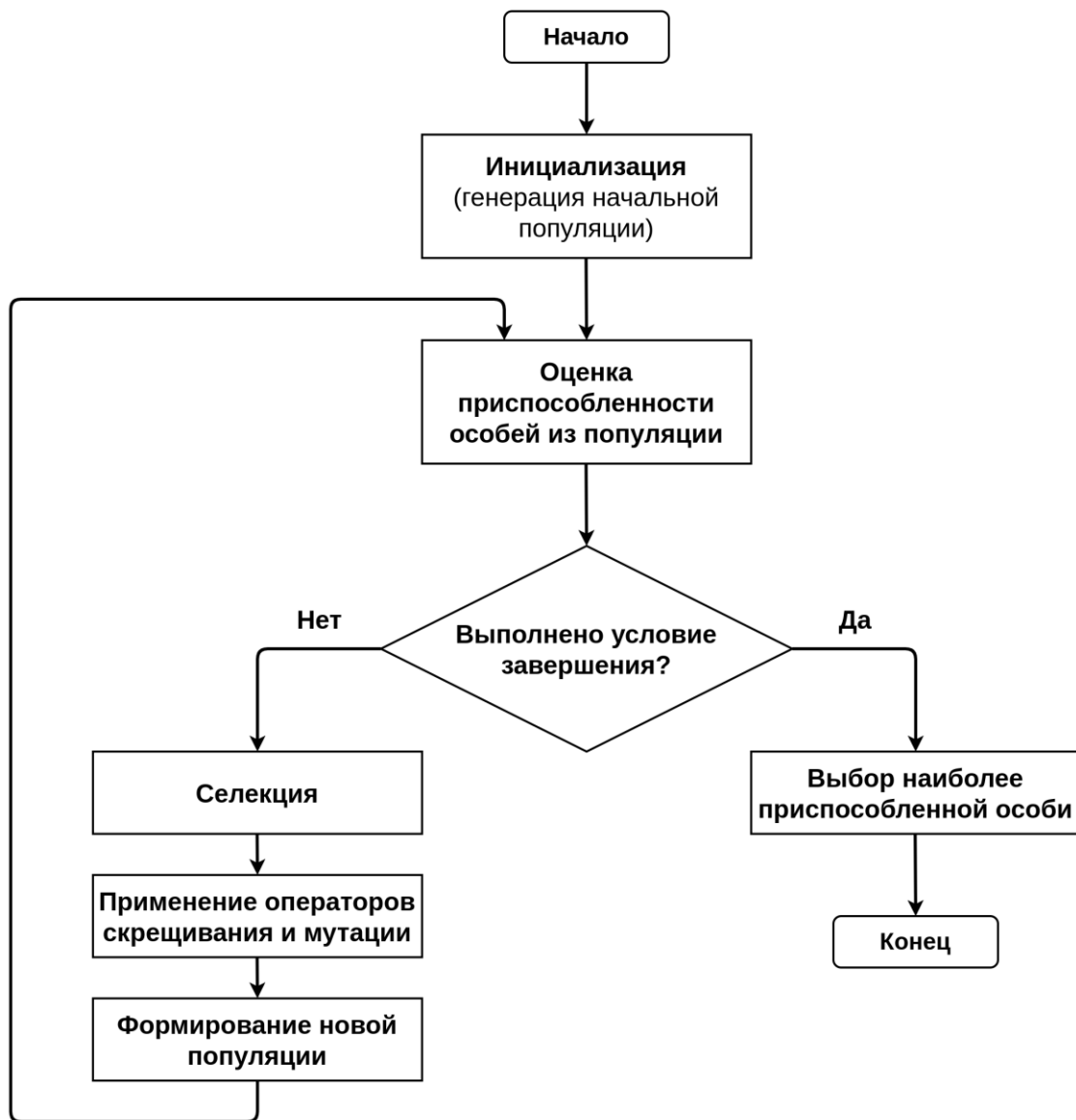


Рис. 1. Блок-схема классического генетического алгоритма.

Классический генетический алгоритм состоит из следующих шагов (рис. 1) [3].

1. Инициализация исходной популяции.
2. Оценивание приспособленности.

3. Проверка условия остановки алгоритма.
4. Селекция хромосом.
5. Применение генетических операторов.
6. Формирование новой популяции.
7. Выбор «наилучшей» хромосомы.

Инициализация исходной популяции, как правило, заключается просто в создании набора особей путём генерации случайным образом значений генов в их хромосомах. Если условия задачи накладывают какие-то ограничения на структуру хромосомы – эти ограничения должны учитываться при генерации исходной популяции.

Оценивание приспособленности заключается в вычислении значения функции приспособленности для каждой особи из популяции. Затем происходит проверка условия остановки алгоритма. Чаще всего условием остановки алгоритма является достижение одной из особей некоторого порогового значения приспособленности (иными словами – нахождение приемлемого решения) или достижение максимального количества поколений.

Если условие остановки выполнено, то алгоритм завершает свою работу и в качестве решения задачи возвращает наиболее приспособленную особь. В противном случае выполняется переход к следующему поколению путём выполнения селекции и применения генетических операторов.

Методы селекции

Селекция заключается в выборе особей, которые будут давать потомство для следующего поколения. Селекция реализует принцип естественного от-

бора, согласно которому более приспособленные особи имеют больше шансов на создание потомства. В результате селекции создаётся так называемая **родительская популяция** или **родительский пул**.

Метод рулетки (англ. roulette wheel selection)

Каждой особи сопоставляется сектор колеса рулетки, размер которого пропорционален значению приспособленности данной особи (рис. 2). Размер сектора вычисляется по формуле

$$v(ch_i) = p(ch_i) \cdot 100\%,$$

где ch_i – хромосома i -й особи, $p(ch_i)$ – вероятность селекции хромосомы ch_i .

Вероятность селекции вычисляется по формуле

$$p(ch_i) = \frac{f(ch_i)}{\sum_{j=1}^n f(ch_j)},$$

где f – функция приспособленности, n – количество особей в популяции.

Для формирования родительского пула производится «вращение» колеса рулетки, и выбирается особь, соответствующая выпавшему сектору. «Вращение» заключается в генерации случайного числа из отрезка $[0,100]$ и определении, в какой «сектор» попало это число. Таким образом, так как размер сектора пропорционален приспособленности особи, в родительский пул с большей вероятностью будут попадать наиболее приспособленные особи.

В методе рулетки предполагается, что решается задача максимизации, то есть ищется максимум функции приспособленности. На практике, напротив, чаще встречаются задачи минимизации. Чтобы свести задачу минимизации некоторой функции к задаче максимизации функции приспособленности, её можно определить как $f(ch_i) = \frac{1}{1+d(ch_i)}$ (если d – некоторая неотрицательная функция, которую нужно минимизировать).

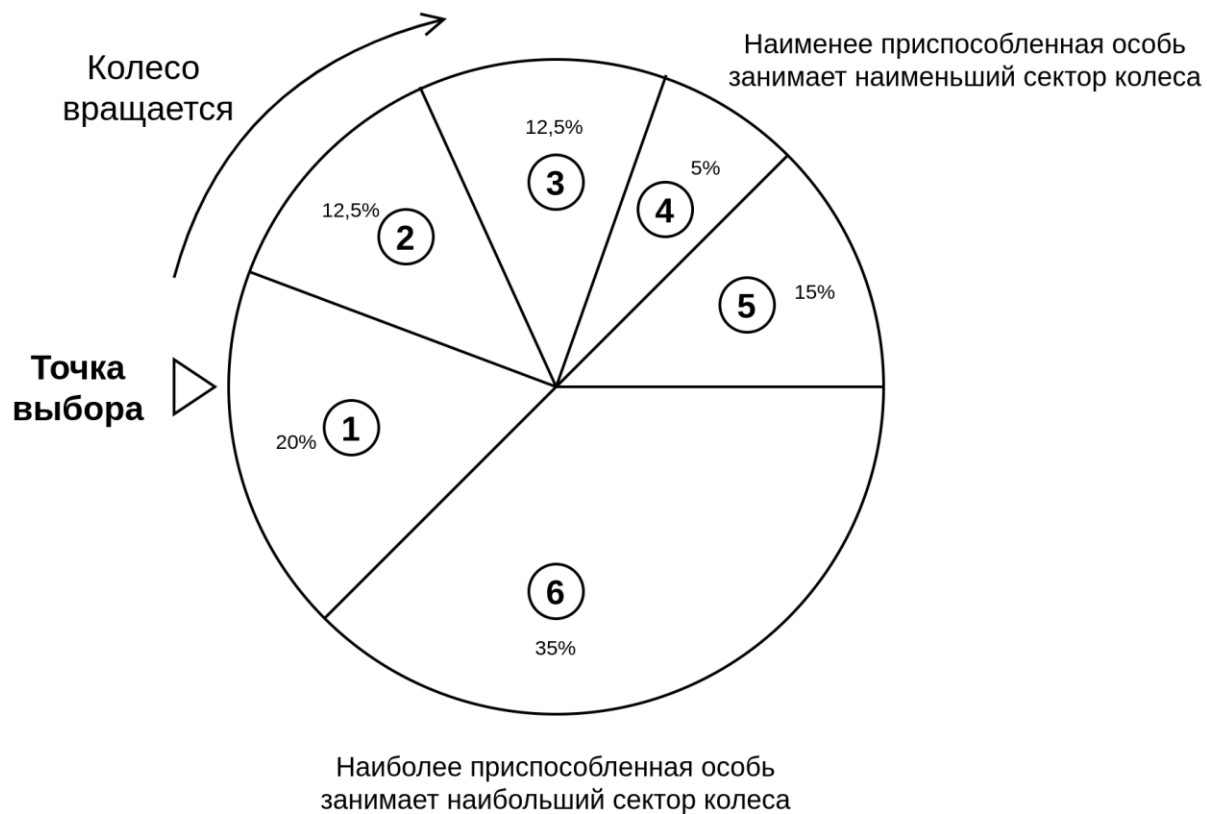


Рис. 2. Иллюстрация разбиения колеса рулетки на секторы.

Турнирный метод (англ. **tournament selection**)

Все особи популяции разбиваются на группы и в каждой группе проводится «турнир», и выбирается наиболее приспособленная особь из данной группы. Эта особь попадает в родительский пул.

Группы могут иметь произвольный размер, но чаще всего это 2-3 особи. Параметр размера группы позволяет регулировать **давление отбора**. Чем меньше размер групп – тем выше шансы «слабых» особей попасть в родительский пул.

Турнирный метод относят к **непараметрическим** методам селекции, так как он нечувствителен к абсолютным значениям функции приспособленно-

сти, и это позволяет ему избежать некоторых проблем, присущих методу колеса рулетки [3]:

- если одна из особей имеет значительно бóльшее значение функции приспособленности, чем другие особи, – это приведёт к вырождению популяции;
- если значения приспособленности меняются незначительно в пределах популяции, селекция превращается в случайный выбор.

Ранговая селекция (англ. ranking selection)

Особи популяции ранжируются в порядке убывания значения функции приспособленности. Каждой особи присваивается **ранг** – число, определяющее её позицию в отсортированном списке. Сколько копий каждой особи попадёт в родительский пул определяется по априорно заданной функции, зависящей от ранга. Этот метод также относится к непараметрическим методам селекции.

Операторы скрещивания

Скрещивание (или **кроссинговер**) предназначено для реализации наследственной изменчивости. Из родительского пула случайным образом формируются пары особей для скрещивания. К сформированным парам применяется оператор скрещивания, в результате чего образуется два (как правило) потомка, каждый из которых наследует часть генов от каждого из родителей. Так как родительский пул был сформирован из наиболее приспособленных особей, то среднее значение приспособленности следующего поколения должно возрасти.

В классическом генетическом алгоритме используется **одноточечный оператор** скрещивания (рис. 3). Он работает следующим образом.

- Случайным образом выбирается **точка скрещивания** l_k , которая представляет собой натуральное число из интервала $[1, L - 1]$, где L – длина хромосомы.
- Хромосома первого потомка на позициях от 1 до l_k содержит гены первого родителя, а на позициях от $l_k + 1$ до $L - 1$ – второго родителя.
- Хромосома второго потомка наоборот – на позициях от 1 до l_k содержит гены второго родителя, а на позициях от $l_k + 1$ до $L - 1$ – первого.



Рис. 3. Пример применения оператора одноточечного скрещивания.

Оператор **двухточечного** скрещивания работает аналогично за исключением того, что используются две точки скрещивания (рис. 4).

| | | | | | | | | | | |
|------------|-------------------|---|---|---|---|---|---|---|---|---|
| | Точки скрещивания | | | | | | | | | |
| Родитель 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 |
| Родитель 2 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| Потомок 1 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| Потомок 2 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |

Рис. 4. Пример применения оператора двухточечного скрещивания.

Также существует **однородный** оператор скрещивания, при использовании которого выбор гена на каждую позицию в хромосоме потомка осуществляется случайно – равновероятно выбирается ген одного из родителей.

Существуют и другие операторы скрещивания, учитывающие особенности конкретных задач. Например, в некоторых задачах гены связаны между собой (иногда такое взаимодействие называют *эпистазом* [2]), и высокое значение функции приспособленности может быть обусловлено удачной комбинацией значений нескольких генов. В этом случае вероятность того, что од-

нородное скрещивание ухудшит потомка, гораздо выше, чем в ситуации, когда гены не зависят друг от друга.

Необходимо учитывать ограничения операторов скрещивания. В частности, если рассматривать хромосомы как точки в L -мерном пространстве, то область пространства, в которую могут попасть потомки, ограничена гиперкубом, в противоположных вершинах которого расположены родительские особи, а потомки могут оказаться в любой из его вершин (рис. 5).

Аналогично, для всей популяции можно построить гиперкуб, ограничивающий область возможных потомков. Поэтому важно при инициализации начальной популяции обеспечить генетическое разнообразие особей, чтобы этот гиперкуб покрывал как можно бóльшую область пространства поиска. Есть высокий риск **преждевременной сходимости** или **вырождения популяции**, если это условие не выполнено [2].

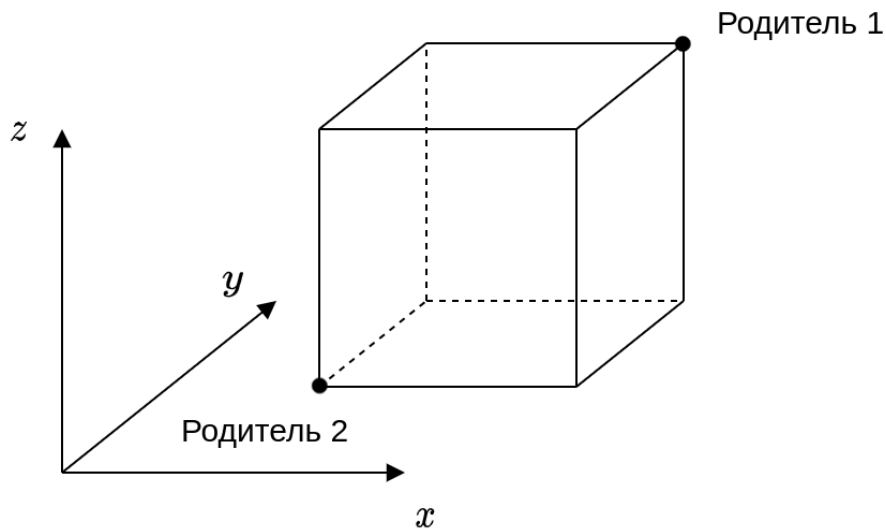


Рис. 5. Куб, ограничивающий возможное положение потомков в пространстве поиска (в случае трёхмерного пространства).

Операторы мутации

Операторы мутации используются для того, чтобы выбраться за пределы упомянутого ранее гиперкуба и привести в популяцию новые аллели генов.

По аналогии с живыми организмами, мутации хромосом в генетических алгоритмах происходят достаточно редко. Частота мутаций регулируется параметром p_m – вероятностью мутации. В классическом генетическом алгоритме используется двоичное кодирование параметров задачи, поэтому применение оператора мутации сводится к изменению значения гена на противоположное с заданной вероятностью.

Если используется не двоичное представление хромосом, то оператор мутации может быть реализован как обмен значениями двух случайных генов или замена значения гена на случайное. Так же, как и при выборе оператора скрещивания, при выборе оператора мутации должны учитываться особенности решаемой задачи.

Пример решения задачи с помощью ГА

Для иллюстрации работы генетического алгоритма рассмотрим пример решения задачи нахождения хромосомы с максимальным количеством единиц. Пусть длина хромосом будет равна 10, а количество особей в популяции – 8.

Инициализация начальной популяции

Для инициализации начальной популяции воспользуемся генератором случайных чисел и сгенерируем 8 хромосом.

$$ch_1 = 0111000101 \quad ch_5 = 1110101101$$

$$ch_2 = 0111101111 \quad ch_6 = 0010111111$$

$$ch_3 = 0111111011 \quad ch_7 = 1100000011$$

$$ch_4 = 0010010111 \quad ch_8 = 0010100000$$

Оценка приспособленности

Для данной задачи в качестве функции приспособленности нужно использовать функцию, которая подсчитывает количество единиц в хромосоме.

$$f(ch_1) = 5 \quad f(ch_5) = 7$$

$$f(ch_2) = 8 \quad f(ch_6) = 7$$

$$f(ch_3) = 8 \quad f(ch_7) = 4$$

$$f(ch_4) = 5 \quad f(ch_8) = 2$$

Среднее значение приспособленности 5,75.

Селекция

Так как в начальной популяции нет хромосомы, содержащей на всех позициях единицы (очевидно, что это решение задачи), то нужно произвести селекцию с целью формирования следующего поколения. Воспользуемся методом колеса рулетки. Для этого рассчитаем величину сектора колеса для каждой хромосомы.

$$v(ch_1) = 11 \quad v(ch_5) = 15$$

$$v(ch_2) = 17 \quad v(ch_6) = 15$$

$$v(ch_3) = 17 \quad v(ch_7) = 9$$

$$v(ch_4) = 11 \quad v(ch_8) = 5$$

«Поворот» колеса рулетки в данном случае эквивалентен генерации случайного числа из отрезка $[0,100]$. Сгенерируем 8 чисел. Предположим, что сгенерированы следующие числа:

$$97, 20, 53, 3, 81, 40, 83, 71.$$

Это соответствует выбору следующих хромосом:

$$ch_8, ch_2, ch_4, ch_1, ch_6, ch_3, ch_6, ch_5.$$

Эти особи попадут в родительский пул.

Применение генетических операторов

Для применения генетических операторов нужно сформировать пары. Это можно сделать случайным образом. Предположим, что были получены следующие пары.

$$ch_1 \text{ и } ch_3, ch_2 \text{ и } ch_4, ch_6 \text{ и } ch_8, ch_6 \text{ и } ch_5.$$

Используем одноточечное скрещивание. Для каждой пары разыграем точку скрещивания: 4,8,6,9. Теперь можно применить оператор скрещивания к сформированным парам.

$$\begin{array}{lcl} ch_1 = 0111000101 & \xrightarrow[l_k=4]{\text{Скрещивание}} & ch_1 = 0111111011 \\ ch_3 = 0111111011 & & ch_2 = 0111000101 \end{array}$$

$$\begin{array}{lcl} ch_2 = 0111101111 & \xrightarrow[l_k=8]{\text{Скрещивание}} & ch_3 = 0111101111 \\ ch_4 = 0010010111 & & ch_4 = 0010010111 \end{array}$$

$$\begin{array}{lcl} ch_6 = 0010111111 & \xrightarrow[l_k=6]{\text{Скрещивание}} & ch_5 = 0010110000 \\ ch_8 = 0010100000 & & ch_6 = 0010101111 \end{array}$$

$$\begin{array}{lcl} ch_6 = 1110101101 & \xrightarrow[l_k=9]{\text{Скрещивание}} & ch_7 = 1110101101 \\ ch_5 = 0010111111 & & ch_8 = 0010111111 \end{array}$$

Установим вероятность мутации $p_m = 0,1$ и применим оператор мутации к каждой из получившихся хромосом-потомков.

$$\begin{array}{lcl} 0111111011 & \longrightarrow & 0111111011 \\ 0111000101 & \longrightarrow & 01\mathbf{0}1100101 \\ 0111101111 & \longrightarrow & 01111011\mathbf{0}1 \\ 0010010111 & \longrightarrow & 0010010111 \\ 0010110000 & \longrightarrow & 001\mathbf{1}110000 \\ 0010101111 & \longrightarrow & 0\mathbf{1}10101111 \\ 1110101101 & \longrightarrow & 1\mathbf{0}10101101 \\ 0010111111 & \longrightarrow & 00101\mathbf{0}1111 \end{array}$$

Среднее значение приспособленности нового поколения составило 5,88.

Теорема схем

Теоретическим обоснованием генетических алгоритмов является *теорема схем*, сформулированная и доказанная Дж. Холландом в 1975 г. Она опирается на понятие «схема».

Под **схемой** понимается некоторый шаблон, позволяющий выделить хромосомы с общими свойствами. Для хромосом, представленных в виде двоичных строк, схема определяется как строка той же длины, что и анализируемые хромосомы, состоящая из символов $\{0,1,*\}$. Символы 0 и 1 определяют значения, которые должны стоять на определённых позициях в хромосоме, а символ * означает «любой символ».

Говорят, что хромосома *соответствует схеме*, если для каждой позиции $j = 1, \dots, L$ значение гена в этой позиции соответствует значению, которое стоит в этой позиции в схеме. Символу * соответствуют как значение 0, так и значение 1.

Например, схеме 10 * соответствуют хромосомы $\{0100, 0101, 1100, 1101\}$.

Для оценки схемы также вводится понятие **приспособленность схемы**, которая определяется как среднее значение функции приспособленности, вычисленное по тем хромосомам популяции, которые соответствуют схеме.

Для анализа схем вводятся понятия *порядок* и *охват* схемы.

Порядок – это количество постоянных позиций в схеме (т. е. символов, отличных от *). Порядок описывает «строгость» схемы. Чем больше порядок схемы, тем меньше хромосом ей соответствует.

Охват (длина) – это расстояние между первым и последним постоянными символами. Охват характеризует «компактность» схемы.

Например, порядок схемы $1 \ast 0$ равен 2, а её охват равен $4 - 1 = 3$.

Понятие «схема» было введено как формальный аппарат для анализа свойств хромосом. Целью анализа было проследить, каким образом меняются свойства хромосом от поколения к поколению в процессе работы генетического алгоритма.

Введение понятий «порядок» и «охват» позволило проанализировать устойчивость схем к операциям скрещивания и мутации. Так, например, чем больше порядок схемы, тем выше вероятность того, что она будет «сломана» в результате мутации. Аналогично, чем выше охват схемы, тем выше вероятность того, что она будет «сломана» в результате скрещивания.

Теперь, когда введены все необходимые понятия, можно сформулировать саму **теорему схем**, которая заключается в следующем. Схемы малого порядка с малым охватом и с приспособленностью выше средней формируют показательно возрастающее количество своих представителей в последующих поколениях генетического алгоритма [3].

На основе этой теоремы была выдвинута **гипотеза о строительных блоках**, которая утверждает, что генетический алгоритм стремится достичь близкого к оптимальному результата путём комбинирования хороших схем малого порядка и малого охвата («строительных блоков»).

Эвристики

Генетические алгоритмы относятся к классу *эвристических* алгоритмов, то есть они не гарантируют нахождение оптимального решения. Поэтому структура описанного ранее классического генетического алгоритма не является неизменной и допускает введение других эвристик. К таким эвристикам

относятся элитарная стратегия, инбридинг и аутбридинг, островная модель эволюции и др.

Элитарная стратегия

В результате скрещивания и мутации наилучшая особь поколения может не передать свои качества потомкам, и та комбинация генов, которая характеризуется высокой приспособленностью, может потеряться и не перейти в следующее поколение. Чтобы избежать такой ситуации вводится так называемая элитарная стратегия, которая заключается в том, что наилучшая особь популяции переходит в новое поколение без изменений. Также можно сохранять не одну особь, а определённую долю наилучших особей. В этом случае говорят о генетическом алгоритме с *частичной заменой популяции*.

Инбридинг и аутбридинг

Рассмотренная ранее стратегия подбора пар называется **панмиксией** – пары формируются случайным образом. Можно также использовать другие стратегии подбора пар исходя из различных соображений.

При **инбридинге** пары формируются из наиболее близких (в смысле некоторой метрики, например, – расстояния Хэмминга) особей. Это позволяет сосредоточить поиск в локальных областях.

При **аутбридинге** пары формируются из наиболее отличающихся особей. Такая стратегия позволяет предупредить преждевременную сходимость и исследовать новые области пространства поиска.

Островная модель эволюции

Генетические алгоритмы требуют больших вычислительных ресурсов. Один из способов ускорения их работы – островная модель эволюции. Она заключается в том, что параллельно запускаются несколько генетических алгоритмов – каждый со своей популяцией. Иногда эти алгоритмы обмениваются наилучшими особями, чтобы расширить область поиска.

Такая модель эволюции позволяет, во-первых, распараллелить вычисления, а, во-вторых, – положительно сказывается на самом эволюционном процессе. В частности, такая модель является более эффективной в задачах мультимодальной оптимизации, так как в различных популяциях могут быть найдены различные максимумы.

Решение задачи коммивояжёра с помощью ГА

Классическим примером задачи комбинаторной оптимизации, которая может быть решена при помощи генетических алгоритмов, является задача коммивояжёра. Она формулируется следующим образом: необходимо найти маршрут минимальной длины между N городами, проходящий через все города по одному разу и возвращающийся в исходный город. Эта задача сводится к поиску в графе гамильтонова цикла минимальной длины.

Рассмотрим конкретный пример задачи коммивояжера – BERLIN52, взятый из репозитория [TSPLIB](#). В этой задаче необходимо найти кратчайший маршрут, соединяющий 52 точки на карте Берлина.

Для этой задачи известен оптимальный маршрут. Он изображён на рис. 6. Его длина составляет 7544,3659.

Прежде всего необходимо выбрать способ кодирования решения задачи в виде хромосомы. Самый простой вариант – представить хромосому в виде последовательности десятичных чисел – номеров вершин графа. Нет необходимости фиксировать начальную вершину, так как маршрут представляет собой цикл. Таким образом, в таком представлении важны не позиции вершин в хромосоме, а их порядок относительно друг друга.

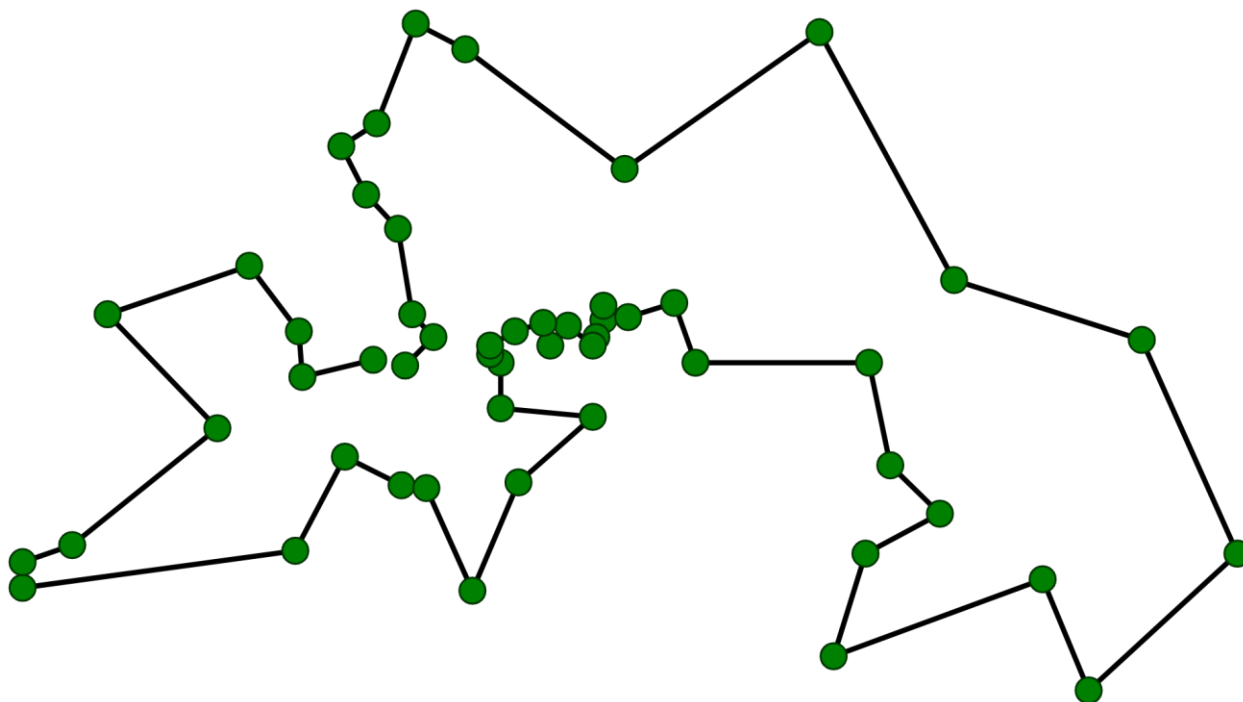


Рис. 6. Оптимальный маршрут для задачи BERLIN52.

Выбранное представление требует определение оператора мутации, учитывающего особенности решаемой задачи. Так как в маршруте не может быть повторяющихся вершин – замена вершины на случайную не подойдёт. Оператор мутации в данном случае можно реализовать как обмен значениями двух случайных позиций в хромосоме.

По тем же причинам нельзя использовать классические операторы скрещивания, – в результате их применения могут образоваться недопустимые маршруты.

Один из вариантов скрещивания, применимый к решаемой задаче, – это *перестановочный* (англ. permutation crossover). Ниже приведён пример применения этого оператора.

| | | |
|--|------------------------------|------------------------------|
| Родитель ₁ = 847 3625 190 | Скрещивание \Rightarrow | Потомок = 0473625 189 |
| Родитель ₂ = 0 1234 56789 | | |

Аналогично двухточечному оператору случайным образом выбираются две позиции l_1 и l_2 в хромосоме первого родителя. Затем гены первого родителя между этих точек копируются в хромосому потомка. Оставшиеся позиции заполняются генами второго родителя, которые ещё не вошли в хромосому потомка, в том порядке, в котором они встречаются в хромосоме второго родителя.

Ещё один возможный вариант – это *жадный* оператор скрещивания [1]. Он реализуется следующим образом.

- **Шаг 1.** Выбирается случайная вершина и записывается в первую позицию хромосомы потомка.
- **Шаг 2** Сравниваются пути, которые ведут из этой вершины в хромосомах родителей, и выбирается кратчайший путь. Соответствующая вершина добавляется к хромосоме потомка. Если обе вершины уже присутствуют в маршруте, то добавляется случайная вершина.
- **Шаг 3** Шаг два повторяется до тех пор, пока полностью не заполнится хромосома потомка.

На рис. 7 представлен наилучший маршрут, который удалось получить с помощью генетического алгоритма, а на рис. 8 – график зависимости значения функции приспособленности от номера поколения. Длина полученного пути составляет 7914,29, что лишь на 5% больше длины оптимального пути.

Данный результат был получен при использовании следующих значений параметров: размер популяции 100, максимальное количество поколений 1000, вероятность мутации 0,5, селекция методом рулетки, жадный оператор скрещивания, элитизм 0,2.

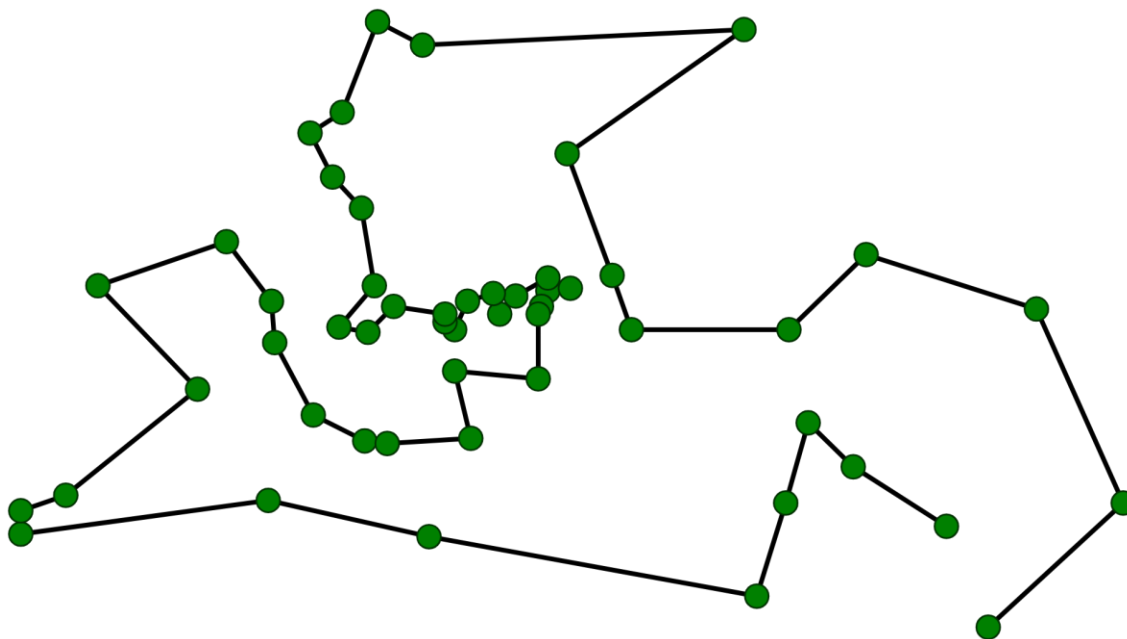


Рис. 7. Маршрут для задачи BERLIN52, полученный с помощью генетического алгоритма.

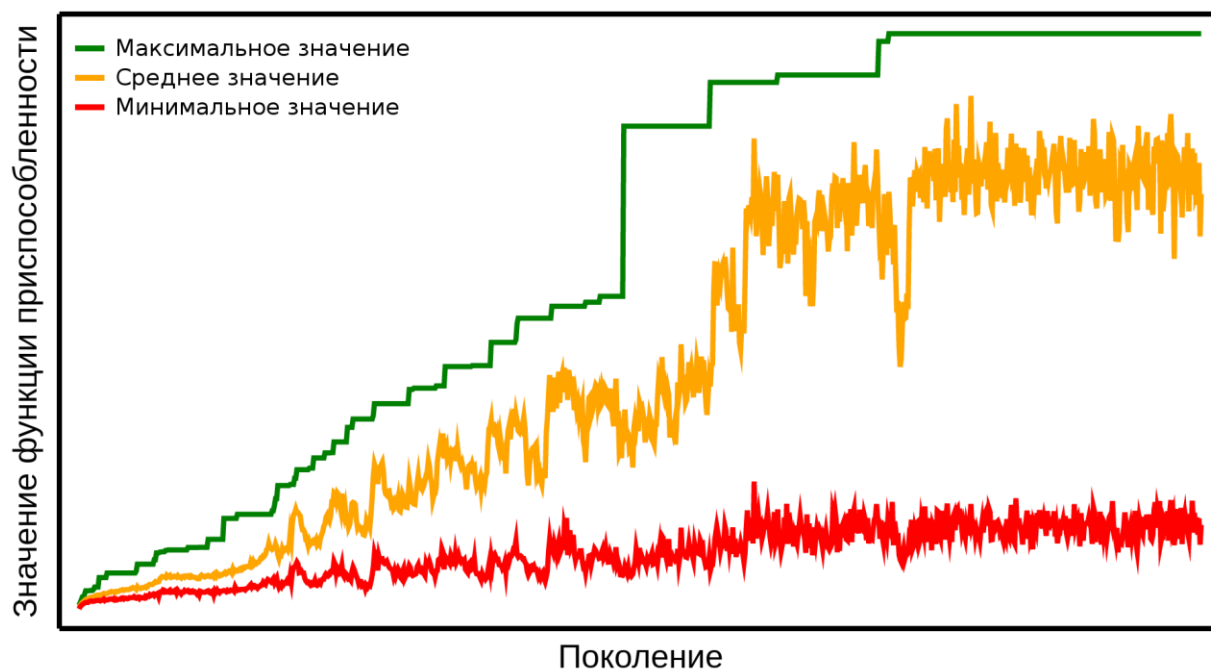


Рис. 8. График, иллюстрирующий зависимость значения функции приспособленности от номера поколения генетического алгоритма.

Особенности генетических алгоритмов

Подводя итог, можно выделить основные особенности генетических алгоритмов по сравнению с другими методами оптимизации.

1. Кодирование параметров.
2. Операции на популяциях.
3. Использование минимума информации о задаче.
4. Рандомизация операций.

Нейроэволюция

Нейроэволюция — один из подходов к решению задачи обучения искусственных нейронных сетей, который заключается в использовании эволюционных алгоритмов для настройки параметров нейронной сети.

Идея использовать эволюционные алгоритмы для обучения искусственных нейронных сетей вполне естественна, ведь мозг человека — прообраз ИНС — является продуктом эволюции. Можно выделить несколько основных преимуществ нейроэволюции перед градиентными методами оптимизации.

Во-первых, эволюционные алгоритмы работают не с одним решением (как градиентный спуск), а с популяцией решений. Это снижает вероятность, что алгоритм сойдётся к ближайшему локальному минимуму, и даёт возможность получить более хорошее решение.

Во-вторых, для работы эволюционных алгоритмов не требуется дифференцируемость целевой функции. Это позволяет им решать задачи, для которых сложно задать дифференцируемую целевую функцию. Например, к таким задачам можно отнести обучение интеллектуальных агентов. Нейроэво-

люция является конкурентоспособной альтернативой градиентным методам оптимизации для задач обучения с подкреплением [10].

Кроме этого, эволюционные алгоритмы можно применять не только для настройки параметров нейронной сети, но и для подбора оптимальной архитектуры нейронной сети (neural architecture search, NAS) и настройки гиперпараметров.

Применение генетических алгоритмов для обучения ИНС

В случае задачи обучения ИНС при помощи генетических алгоритмов решением задачи (т. е. особью) будет набор значений синаптических весов и смещений нейронов. Для задания генотипа особи необходимо определить способ хромосомного кодирования.

Так как значения синаптических весов и смещений представляют собой произвольные действительные числа, возникают определённые сложности с их кодированием. Один из возможных вариантов кодирования заключается в следующем. Можно ограничить диапазон изменения весов $[a, b]$, а также задать точность их представления k знаков после запятой. Тогда длина двоичной последовательности (l) для представления одного параметра должна удовлетворять неравенству

$$(b - a) \cdot 10^k \leq 2^l - 1.$$

Таким образом, каждый параметр нейронной сети можно представить в виде двоичной последовательности. Каждой такой последовательности соответствует десятичное число, но это число не является значением параметра. Чтобы перейти от этого числа к значению параметра, необходимо воспользоваться формулой

$$w = a + d_w \frac{b - a}{2^l - 1},$$

где w — значение параметра нейронной сети, a и b — левая и правая границы диапазона допустимых значений параметра, l — длина двоичного представления, d_w — десятичное значение двоичной последовательности, кодирующей параметр w .

Чтобы не ограничивать точность представления значений, допускается использование действительных чисел, однако, такой вариант представления выходит за рамки классических генетических алгоритмов, в которых используются двоичные последовательности [12].

Определив способ хромосомного кодирования, можно инициализировать начальную популяцию. Следующий шаг — определить функцию приспособленности. Выбор функции приспособленности зависит от типа решаемой задачи. Например, для задачи классификации функцией приспособленности может служить точность (англ. accuracy). Для других задач также могут использоваться соответствующие им метрики.

После определения функции приспособленности становится возможным провести оценку приспособленности особей популяции, и на основе этого сформировать родительский пул при помощи одного из методов селекции (например, метода рулетки или турнирного метода).

Реализация генетических операторов зависит от выбранного способа хромосомного кодирования. Если используется двоичное представление, можно воспользоваться классическими реализациями операторов мутации (инверсия значения) и скрещивания (одноточечный, двухточечный, однородный, и т. д.). Описанный ранее способ кодирования гарантирует, что любая двоичная строка заданной длины будет представлять допустимое значение параметра.

Если используется представление параметров в виде действительных чисел, оператор мутации можно реализовать как прибавление к значению гена случайного значения из стандартного нормального распределения. Формально это можно выразить как

$$w_q = w_q + \gamma \cdot \xi, \quad \xi \sim \mathcal{N}(0,1),$$

где w_q — значение мутируемого гена, ξ — стандартная нормальная случайная величина, γ — некоторая константа, значение которой можно подбирать эмпирически.

При реализации оператора скрещивания возникает гораздо больше трудностей, чем при реализации оператора мутации. В результате скрещивания потомки должны перенять часть свойств от каждого из родителей. При этом нужно учитывать, что структурной единицей нейронной сети является нейрон, поэтому при скрещивании имеет смысл обрабатывать каждый нейрон по отдельности.

В простейшем случае — если нейронная сеть состоит лишь из одного нейрона — реализация оператора скрещивания тривиальна. В этом случае хромосома может представлять собой вектор весов нейрона, и к ней можно применить один из стандартных операторов скрещивания.

В случае обучения многослойной сети при помощи ГА возникает так называемая **проблема перестановки** (англ. *permutation problem*), которая должна учитываться при реализации оператора скрещивания. Предположим, имеется двухслойная полносвязанная нейронная сеть прямого распространения. Можно заметить, что если переставить местами нейроны в скрытом слое, то это никак не отразится на работе нейронной сети. Иными словами, такая нейронная сеть *инвариантна* относительно операции перестановки нейронов в скрытом слое.

Геном многослойной сети можно представить в виде одной длинной хромосомы, в которой будут содержаться параметры всех нейронов, либо в виде совокупности хромосом — по одной на каждый нейрон. Предположим, что используется второй вариант. Тогда необходимо, чтобы при скрещивании своими участками обменивались хромосомы, соответствующие «эквивалентным» нейронам в родительских особях. В противном случае, операция скрещивания может препятствовать эволюционному процессу, разрушая полученные ранее комбинации синаптических весов.

Ввиду инвариантности нейронной сети к перестановке нейронов в скрытом слое, нейроны, отвечающие за выделение из данных одних и тех же (схожих) признаков, могут оказаться на разных позициях в скрещиваемых особях. В этом и заключается проблема перестановки — нельзя просто скрещивать между собой хромосомы нейронов, стоящих на одинаковых позициях в родительских особях, а нужно искать «эквивалентные» нейроны (иными словами функционально схожие нейроны) и скрещивать их друг с другом. На рис. 9 представлена иллюстрация проблемы перестановки.

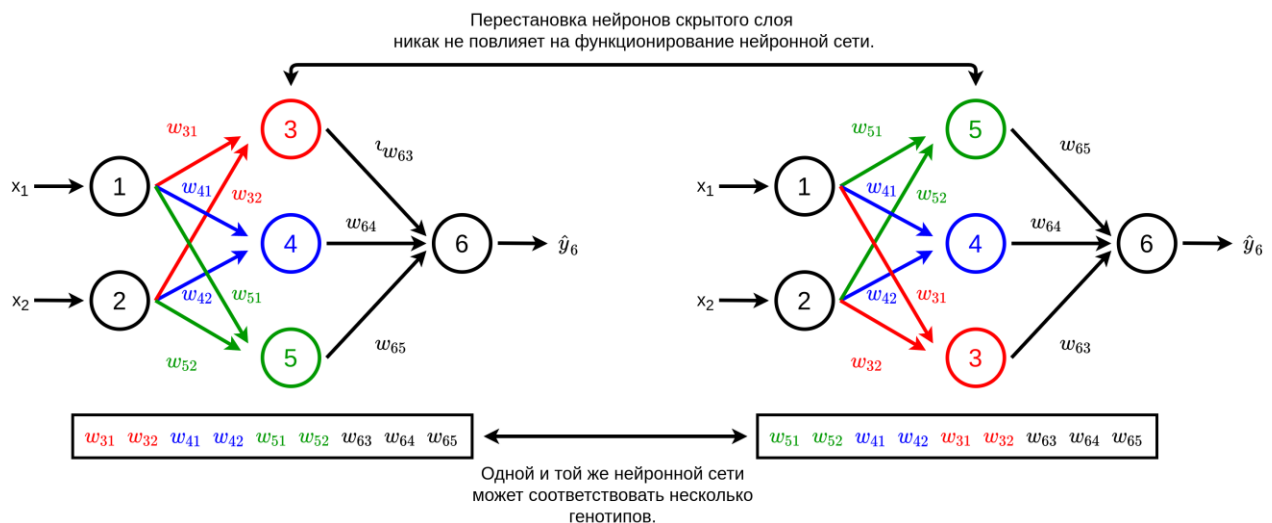


Рис. 9. Иллюстрация проблемы перестановки.

Самый простой способ решить проблему перестановки — отказаться от оператора скрещивания и оставить только мутации. Более правильный вариант — использовать специальные операторы скрещивания, учитывающие эту проблему [9,11].

После применения генетических операторов остается только сформировать новую популяцию и перейти на следующую итерацию генетического алгоритма. Эволюционный процесс можно остановить, как только будет получено приемлемое решение, либо если алгоритм достигнет максимального количества итераций (поколений).

Вопросы для самопроверки

1. В чём заключается главная идея эволюционного моделирования?
2. Дайте определения основных понятий, используемых в генетических алгоритмах (популяция, особь, хромосома, генотип, ген, фенотип).
3. Перечислите основные шаги классического генетического алгоритма.
4. В чём заключается процесс селекции? Расскажите, как работают метод рулетки и турнирный метод.
5. Что такое оператор скрещивания? Расскажите, как работают одноточечный, двухточечный и однородный операторы скрещивания.
6. Что такое оператор мутации? Расскажите, какие операторы мутации вы знаете.
7. В чём состоит смысл теоремы схем и гипотезы о строительных блоках?

8. Какими особенностями обладают генетические алгоритмы по сравнению другими методами оптимизации?
9. Какие способы хромосомного кодирования можно использовать при решении задачи обучения нейронной сети с помощью генетического алгоритма?
10. В чём заключается проблема перестановки?

Список литературы

1. Goldberg D. E. Genetic algorithms in search, optimization, and machine learning. MA: Addison-Wesley, 1989.
2. [Sean Luke Essentials of Metaheuristics. Lulu, 2013.](#)
3. Рутковская Д., Пилиньский М., Рутковский Л. Нейронные сети, генетические алгоритмы и нечеткие системы: Пер. с польск. И. Д. Рудинского. М.: Горячая линия - Телеком, 2006. 452 с.
4. [Mauro Dragoni, Antonia Azzini, Andrea G. B. Tettamanzi SimBa: A novel similarity-based crossover for neuro-evolution // Neurocomputing. 2014. Vol. 130. P. 108–122.](#)
5. [Stanley, K.O., Clune, J., Lehman, J. et al. Designing neural networks through neuroevolution // Nature Machine Intelligence. 2019. № 1. P. 24–35.](#)
6. [Thomas Uriot, Dario Izzo Safe Crossover of Neural Networks Through Neuron Alignment // In Proceedings of the 2020 Genetic and Evolutionary Computation Conference \(GECCO '20\). Association for Computing Machinery, New York, NY, USA. 2020. P. 435–443.](#)