

Лекция

Эволюционные алгоритмы.
Генетические алгоритмы.

Эволюционные алгоритмы

- Эволюционные алгоритмы основаны на принципе «выживает наиболее приспособленный».
- Базовые принципы:
- «Популяция», состоящая из большого количества потенциальных решений.
 - Случайные изменения в решениях.
 - Отбор решений в следующее поколение на основе *приспособленности*. Приспособленность отражает качество решения.
- Т.е. ЭА — вариант случайного поиска, но вместо одного текущего решения — популяция.

- Достоинства ЭА
 - Универсальный механизм для решения большого класса оптимизационных задач
 - Применимы для слабо формализованных задач
 - Применимы для задач с большим пространством решений
 - Достаточно легко распараллеливаются
- Недостатки ЭА
 - Нет гарантий качества результата
 - Универсальность => не учитываются особенности конкретной задачи
 - Обычно требуют значительных вычислительных затрат.

Генетические алгоритмы

- Наиболее известная и широко используемая разновидность эволюционных вычислений.

Отличительные особенности:

- Потенциальное решение представляется в виде цепочки символов (обычно бинарных, но возможен и произвольный алфавит) — «хромосома» или «особь». Отдельные символы называются *генами*.
- Для получения следующей популяции к текущей популяции применяются *генетические операторы*: мутация, скрещивание, отбор. Т.е. кроме чисто вероятностных преобразований (мутаций) используются скрещивания - преобразования, «сохраняющие» и «усиливающие» удачные фрагменты решений.

- **Генетические алгоритмы** возникли в результате наблюдения и попыток копирования естественных **процессов, происходящих в мире живых организмов**, в частности, **эволюции и связанной с ней селекции** (естественного отбора) популяций живых существ. Идею генетических алгоритмов (genetic algorithm - GA) высказал **Джон Холланд** в конце 60-х годов XX века. Холланд был уверен в возможности составить и реализовать в виде компьютерной программы алгоритм, который будет **решать сложные задачи** так, как это делает природа — путём **эволюции**.
- Эти алгоритмы имитировали эволюционные процессы в **поколениях таких хромосом**. В них были реализованы механизмы **селекции и репродукции**, аналогичные существующим в природе. Для отражения **приспособленности хромосомы** требовалась некоторая оценка. **Механизм селекции** заключается в выборе хромосом с **наивысшей оценкой** (наиболее приспособленных), которые репродуцируют чаще, чем особи с более низкой оценкой (хуже приспособленные).

- **Репродукция** означает создание новых хромосом в результате рекомбинации генов родительских хромосом.
Рекомбинация — это процесс, в результате которого возникают новые комбинации генов. Для этого используются две операции: **скрещивание**, позволяющее создать две совершенно новые хромосомы потомков путём комбинирования генетического материала пары родителей, **мутация**, которая может вызывать изменения в отдельных хромосомах.
- *Мутация* — случайное изменение одной или нескольких позиций в хромосоме. Например, 1010011 → 1010001.

Разновидностью скрещивания является *Кроссинговер* (кроссовер) — операция, при которой две хромосомы обмениваются своими частями. Например,

1100 & 1010 → 1110 & 1000.

В ГА применяется ряд терминов, заимствованных из генетики, прежде всего: гены и хромосомы, а также популяция, особь, аллель, генотип, фенотип.

Ключевые параметры конкретного ГА:

- *Представление потенциальных решений*
- *Оценка приспособленности (стоимость решения, для максимизации)*
- Операция мутации
- Операция скрещивания
- Операция отбора (селекции)

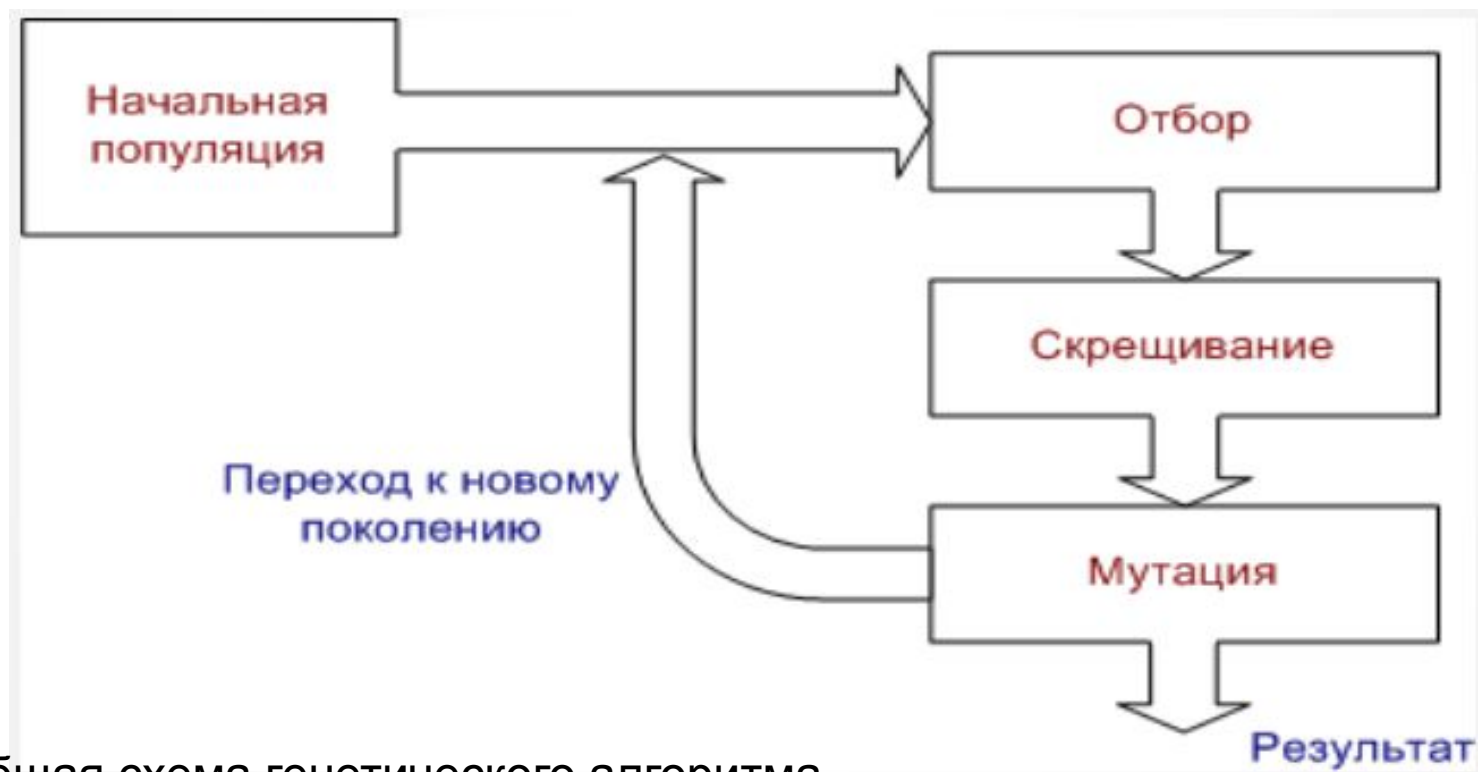


Рисунок 1 – Общая схема генетического алгоритма

Основные понятия

- **Популяция** — конечное множество особей.
Особь, входящие в популяцию, представляются хромосомами с закодированными в них параметрами задачи, т.е. решениями, которые иначе называются **точками в пространстве поиска**.
В некоторых работах особи называются **организмами**.
Хромосомы (цепочки или кодовые последовательности) — это упорядоченные **последовательности генов**.
Ген (свойство, знак или детектор) — атомарный элемент **генотипа**, в частности, хромосомы.

- **Генотип или структура** — это набор хромосом данной особи.

Следовательно, особями популяции могут быть **генотипы** либо **единичные хромосомы** (распространённый случай: генотип состоит из одной хромосомы).

Фенотип — это набор значений, соответствующих данному генотипу, т.е. декодированная структура или множество параметров задачи (решение, точка пространства поиска).

Аллель — совокупность подряд идущих генов

Локус или позиция — позиция гена в хромосоме (цепочке).

Множество позиций генов — это локи.

- Очень важным понятием в теории ГА является **функция приспособленности** (fitness function), иначе называемой **функцией оценки**. Она представляет меру приспособленности данной особи в популяции.
- В задачах оптимизации подобная функция называется целевой функцией и максимизируется или минимизируется в зависимости от условий задачи.
- При помощи функции приспособленности оценивается приспособленность каждой особи данной популяции на каждой итерации ГА, на основе чего создаётся следующая популяция особей, составляющих множество потенциальных решений проблемы, например, задачи оптимизации.
- Очередная популяция в ГА называется **поколением**, а к вновь создаваемой популяции особей применяется термин «новое поколение» или «поколение потомков».

Пример представления популяций:

Рассмотрим функцию

- $f(x) = 2x^2 + 1$

и допустим, что принимает значения из интервала от 0 до 15.

Задача оптимизации этой функции заключается в перемещении по пространству, состоящему из 16 точек со значениями 0, 1, ..., 15 для обнаружения той точки, в которой функция принимает максимальное или минимальное значение.

В качестве параметра задачи выступает переменная .

Множество $\{0, 1, \dots, 15\}$ составляет **пространство поиска** и одновременно — множество потенциальных решений задачи.

Каждое из 16 чисел из этого множества называется **точкой пространства поиска, решением, значением параметра, фенотипом**. Решение, оптимизирующее функцию, называется **наилучшим** или **оптимальным** решением.

- Значения параметра x можно закодировать следующим образом:

0000 0001 0010 0011 0101 0110 0111

1000 1001 1011 1100 1101 1110 1111

Эти последовательности также называются **цепями** или **хромосомами**. В данном примере они выступают в роли **генотипов**. Каждая из хромосом состоит из 4 генов.

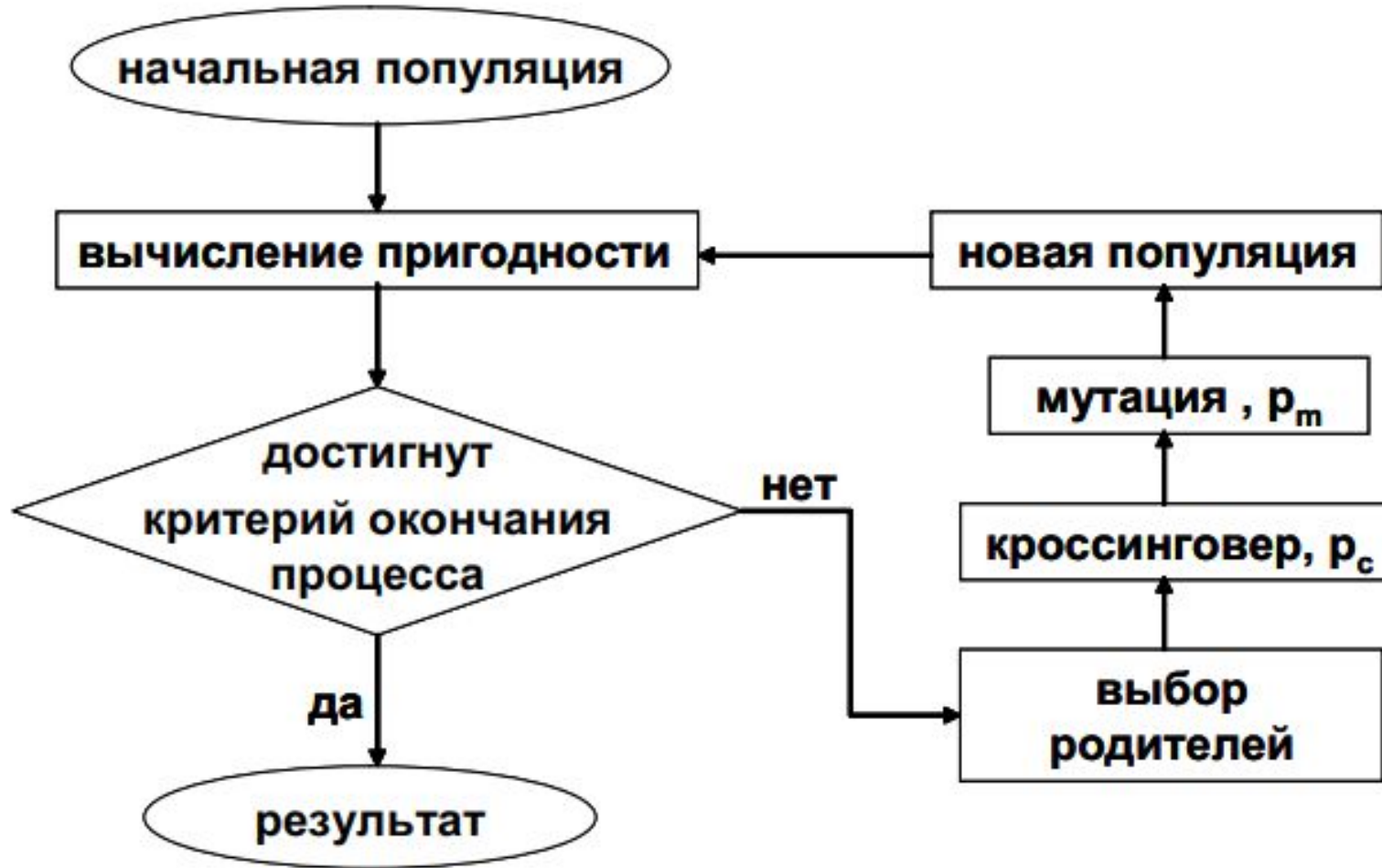
- Значение гена в конкретной позиции называется **аллелью**, принимающей в данном случае значения 0 или 1.
Популяция состоит из особей, выбираемых среди указанных 16 хромосом.
Примером популяции с численностью, равной 6 может быть, например, множество хромосом [0010, 0101, 0111, 1001, 1100, 1110], представляющих собой закодированную форму следующих **фенотипов**: [2, 5, 7, 9, 12, 14]

- **Функция приспособленности** Fitness в этом примере задаётся выражением:

$$f(x)=2x^2 + 1.$$

Приспособленность отдельных хромосом в популяции определяется значением этой функции для значений , соответствующих этим хромосомам, т.е. для **фенотипов**, соответствующих определённым **генотипам**.

Схема простого ГА



• Рисунок 2

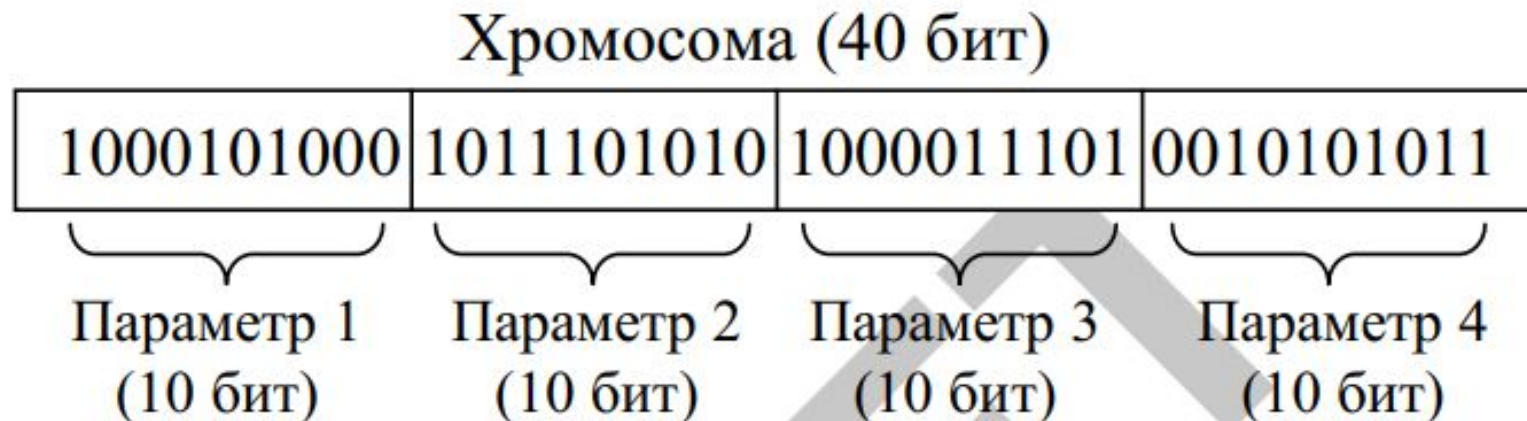
Основные принципы работы ГА

- Основные принципы работы ГА заключены в следующей схеме (см. рис. 2):
 1. Генерируем начальную популяцию из n хромосом.
 2. Вычисляем для каждой хромосомы ее пригодность.
 3. Выбираем пару хромосом-родителей с помощью одного из способов отбора.
 4. Проводим *кроссинговер* двух родителей с вероятностью p_c , производя двух потомков.
 5. Проводим мутацию потомков с вероятностью p_m .
 6. Повторяем шаги 3–5, пока не будет сгенерировано новое поколение популяции, содержащее n хромосом.
 7. Повторяем шаги 2–6, пока не будет достигнут критерий окончания процесса.

Кодирование

- Как правило, в хромосоме кодируются численные параметры решения. Для этого возможно использование целочисленного и вещественного кодирования

Целочисленное кодирование. В каноническом генетическом алгоритме хромосома представляет собой битовую строку, в которой закодированы параметры решения поставленной задачи. Ниже показан пример кодирования 4-х 10-разрядных параметров в 40 разрядной хромосоме. Как правило, считают, что каждому параметру соответствует свой *ген*. Таким образом, можно также сказать, что хромосома на рис. состоит из 4-х 10-разрядных генов.



- Представление числовых значений
Стандартное двоичное представление целых чисел
неудобно для
генетических алгоритмов тем, что небольшое изменение
числового значения может приводить к существенному
изменению строки представления, и
наоборот.

$7 \leftrightarrow 0111$; $8 \leftrightarrow 1000$;
 $15 \leftrightarrow 1111$

Поэтому обычно используется (рефлексивная двоичная)
кодировка Грея.

Коды Грея

Пусть $s = (s_1, s_2, \dots, s_n)$ — стандартное двоичное представление числа,
 $b = (b_1, b_2, \dots, b_n)$ — представление того же числа в кодировке Грея.

Преобразование из стандартного представление в коды Грея:

$$b_k = \begin{cases} s_1, & \text{если } k = 1; \\ s_k \oplus s_{k-1}, & \text{если } k > 1, \quad k = \overline{1, n}. \end{cases}$$

Преобразование из кода Грея в стандартное представление.

$$s_k = \bigoplus_{i=1}^k b_i, \quad k = \overline{1, n},$$

Десятичные числа	Двоичный код	Код Грея	Десятичные числа	Двоичный код	Код Грея
0	0000	0000	8	1000	1100
1	0001	0001	9	1001	1101
2	0010	0011	10	1010	1111
3	0011	0010	11	1011	1110
4	0100	0110	12	1100	1010
5	0101	0111	13	1101	1011
6	0110	0101	14	1110	1001
7	0111	0100	15	1111	1000

Вещественное кодирование.

- Часто бывает удобнее кодировать в гене не целое число, а вещественное. Это позволяет избавиться от операций кодирования/декодирования, используемых в целочисленном кодировании, а также увеличить точность найденного решения. Пример вещественного кодирования представлен ниже.



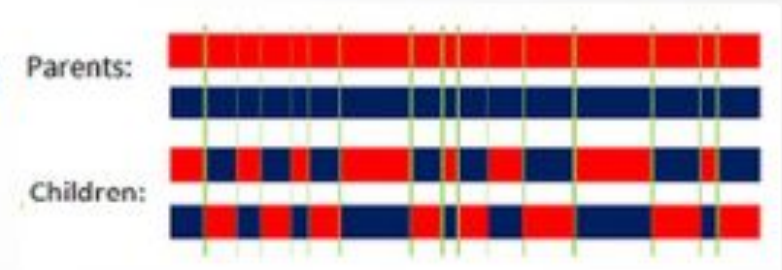
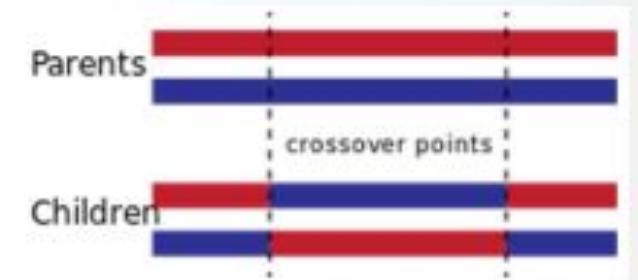
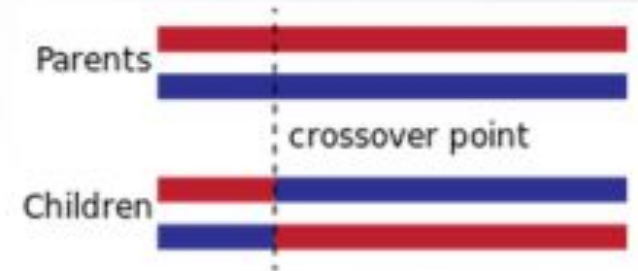
Скрещивание

- Скрещивание (кроссовер, кроссинговер)
Построение новых решений за счёт рекомбинации фрагментов имеющихся решений.
Порядок выполнения:
 1. Выбрать множество родительских хромосом (как правило, 2 шт).
 2. Скрестить их между собой — получить хромосомы нового поколения.Варианты выбора пар хромосом:
 - выбирается k случайных пар из популяции
 - равновероятно;
 - вероятность объединения хромосом в пары зависит от сходства их генотипов (представлений) ρ_g
 - вероятность объединения хромосом в пары зависит от сходства их фенотипов (решений),
 - вся популяция случайно разбивается на пары, каждая пара скрещивается с вероятностью ρ_c

Варианты оператора кроссовера в случае целочисленного кодирования

Варианты оператора скрещивания:

- Одноточечное скрещивание
 - Случайно выбирается позиция разрыва
 - Фрагменты хромосом меняются парами
- Двухточечное скрещивание
- Однородное скрещивание
 - Каждый ген с заданной вероятностью остаётся в «родной» хромосоме или переходит в другую



Скрещивание (кроссовер, кроссинговер)

Для представления в виде перестановки (задача коммивояжера) лучше

подходят операторы скрещивания, старающиеся сохранить относительный порядок элементов. Например — порядковый оператор скрещивания (ОХ):

1) Одну из родительских хромосом считаем «**разрезаемой**», другую

«**заполняемой**». Потом ,для формирования другого потомка, меняем

«роли» хромосом.

2) Случайно выбираем две точки.

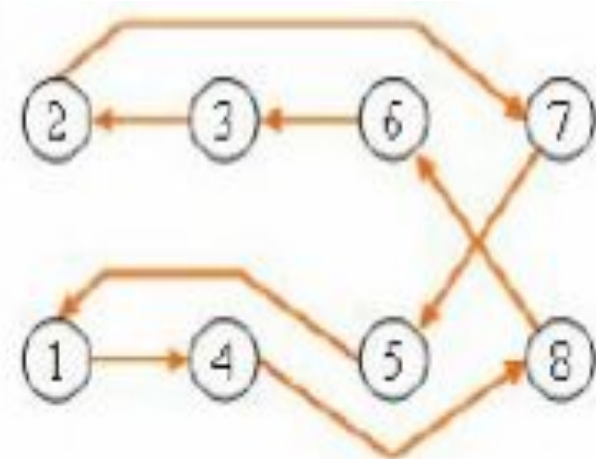
3) Фрагмент разрезаемой хромосомы переносим в потомка на те же позиции.

4) Из заполняемой хромосомы, начиная со второй точки разреза последовательно (и циклически) выбираем элементы, которых нет

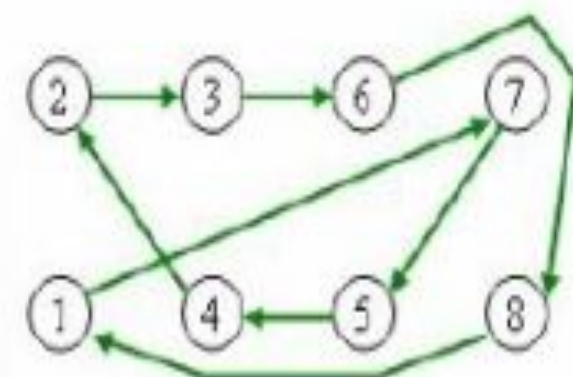
в перенесённом фрагменте (3).

5) Элементы последовательности (4) в том же порядке расставляем в

незанятые позиции потомка, начиная сразу после второй точки разреза



$$s_P^1 = (1, 4, 8, 6, 3, 2, 7, 5) \quad L = 22$$

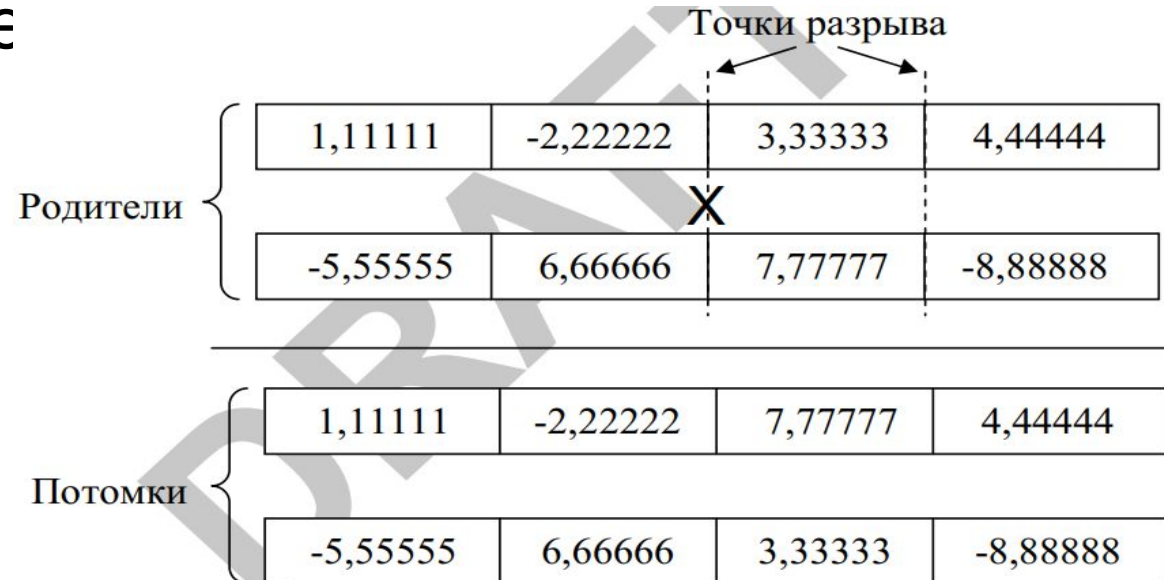


$$s_P^2 = (1, 7, 5, 4, 2, 3, 6, 8) \quad L = 14$$

Варианты оператора кроссовера в случае вещественного кодирования

- **Вещественное кодирование.** Для вещественного кодирования рассмотрим 2-точечный, арифметический и *BLX- α* операторы кроссовера.

2-точечный кроссовер для вещественного кодирования, в целом, аналогичен 2-точечному кроссоверу для целочисленного кодирования. Различие заключается в том, что точка разрыва не может быть выбрана «внутри» гена, а должна попасть ме



При использовании арифметического и $BLX-\alpha$ операторов обмен информацией между родительскими особями и потомками производится с учетом значений генов родителей.

Обозначим $g_k^{(1)}$ и $g_k^{(2)}$ – k -е гены родительских особей, $1 \leq k \leq N$, N – количество генов в хромосоме. Пусть также $h_k^{(1)}$ и $h_k^{(2)}$ – k -е гены потомков. Тогда для арифметического кроссовера:

$$h_k^{(1)} = \lambda g_k^{(1)} + (1 - \lambda) g_k^{(2)},$$

$$h_k^{(2)} = \lambda g_k^{(2)} + (1 - \lambda) g_k^{(1)},$$

где $0 \leq \lambda \leq 1$.

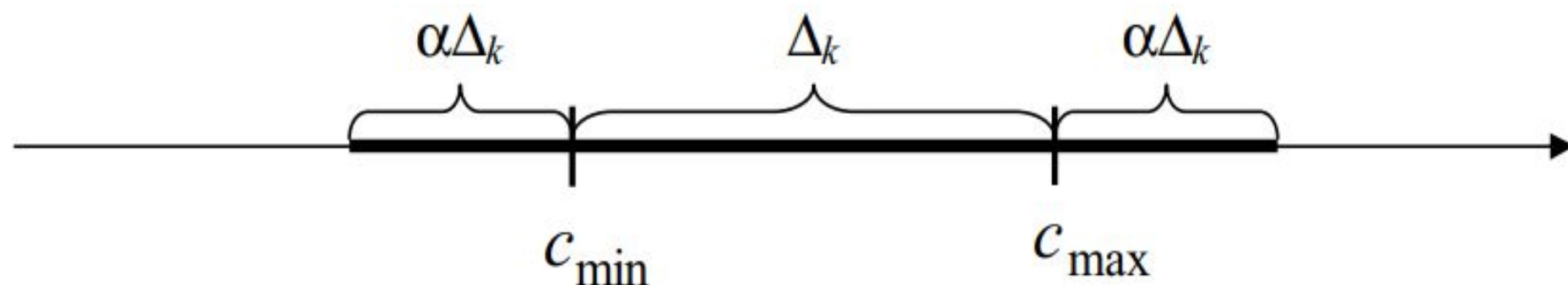
Если используется $BLX-\alpha$ кроссовер, то значение k -го гена потомка выбирается случайным образом (равномерное распределение) из интервала $[c_{\min} - \alpha\Delta_k, c_{\max} + \alpha\Delta_k]$, где α – константа,

$$c_{\min} = \min\{g_k^{(1)}, g_k^{(2)}\},$$

$$c_{\max} = \max\{g_k^{(1)}, g_k^{(2)}\},$$

$$\Delta_k = c_{\max} - c_{\min}.$$

Изображение интервала, используемого для $BLX-\alpha$ кроссовера показано



Мутация

Независимое преобразование отдельных хромосом.

Варианты операторов мутации:

- Случайно выбирается хромосома, в ней случайно выбирается изменяемый ген
- Во всех хромосомах каждый ген меняется с вероятностью pt

(«скорость мутации»). Для двоичного представления: $0 \longleftrightarrow 1$.

$$x_1 x_2 \dots x_{n-1} x_n x_{n+1} \dots x_m \longrightarrow x_1 x_2 \dots x_{n-1} \bar{x}_n x_{n+1} \dots x_m$$

Иногда используются альтернативные варианты мутаций. Для случая, когда хромосома представляет собой перестановку (задача коммивояжёра), мутация тоже может выполняться в виде перестановки:

- Транспозиция случайной пары позиций в хромосоме.
- Сдвиг случайно выбранного гена на случайное количество позиций; промежуточные гены смещаются на 1 позицию в противоположном направлении.
- Инверсия: случайным образом выбирается фрагмент хромосомы и инвертируется (записывается в обратном порядке)

При вещественном кодировании к генам в процессе мутации может добавляться случайный шум.

Отбор

С помощью операции отбора (селекции) формируется новое поколение популяции. Отбор выполняется случайным образом на основе приспособленности каждой хромосомы — более приспособленные хромосомы имеют больше шансов выжить. Классический вариант отбора:

- рассчитать для каждой хромосомы её приспособленность (fitness) f_i
- рассчитать среднее значение приспособленности: f_c
- для каждой хромосомы в новую популяцию попадает:
 - $\text{int}(f_i / f_c)$ копий этой хромосомы
 - ещё одна копия с вероятностью $\{f_i / f_c\}$

- Альтернативные варианты отбора
Используются для предотвращения двух проблем классического отбора: а) преждевременной сходимости (при появлении особей с экстремально высокой приспособленностью), б) потерей чувствительности (при приближении к оптимуму).
 - Ранговый отбор
Аналогичен классическому, но вместо приспособленности хромосомы/особи используется её ранг — порядковый номер в упорядоченной(по приспособленности) последовательности.
 - Турнирный отбор
Для выбора N хромосом проводится N турниров. Для каждого турнира случайно выбираются k хромосом и в новое поколение отбирается наилучшая из них.

Формирование поколений

-

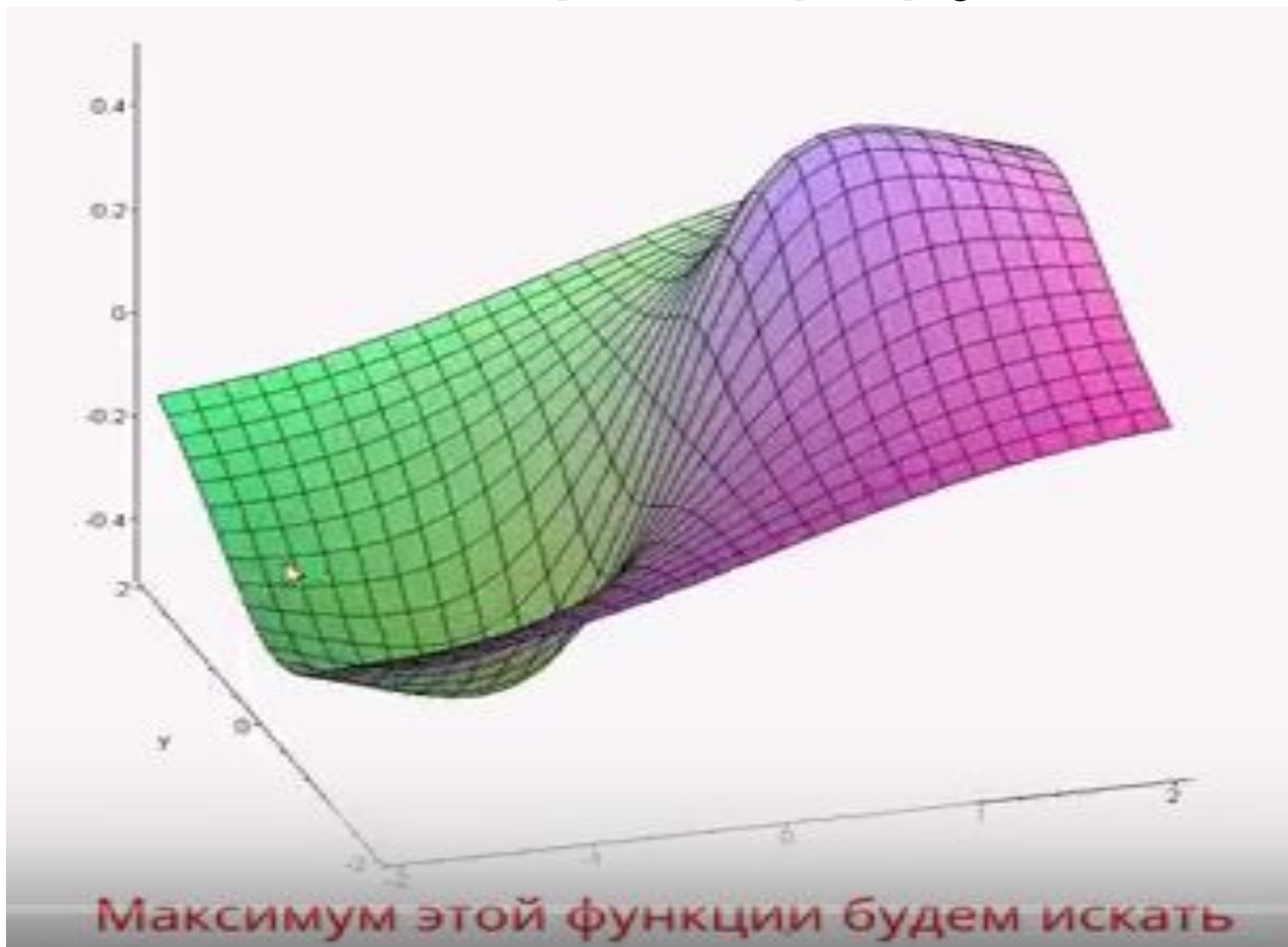
Классический подход: все потомки объединяются вместе с предками в одно множество из $2N$ особей, из этого множества выбирается новое поколение.

Варианты:

- При отборе учитывается «генетическое разнообразие»: из множества особей с одинаковой приспособленностью выбираются максимально отличающиеся по генотипу.
- Элитный отбор: гарантированно выживают k наилучших решений, остальные отбираются каким-то другим способом.

Пример функции

$$z = \frac{x}{x^2 + 2y^2 + 1}$$



4

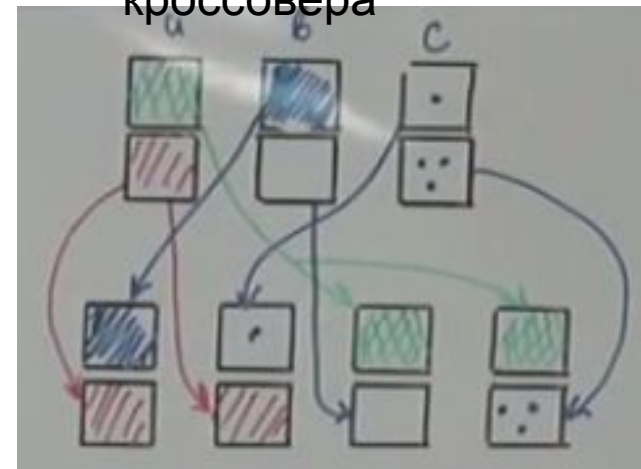
Хромосомы а

$$x = \begin{bmatrix} -2 \\ -1 \\ 0 \\ 2 \end{bmatrix}, \begin{bmatrix} -1 \\ -2 \\ -1 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ -1 \\ -1 \\ 1 \end{bmatrix}, \begin{bmatrix} 2 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

2
гена

$$-0,4 \quad -0,1 \quad 0 \quad 0,286$$

Схема
кроссовера



	c	b	a	
x	-1	0	2	2
y	1	1	-2	-1
	-0,25	0	0,154	0,286

x	0	2	2	2
y	-1	-1	1	-2
	0	0,286	0,286	0,154

x	2	2	2	2
y	1	1	-2	-1
	0,286	0,286	0,154	0,286

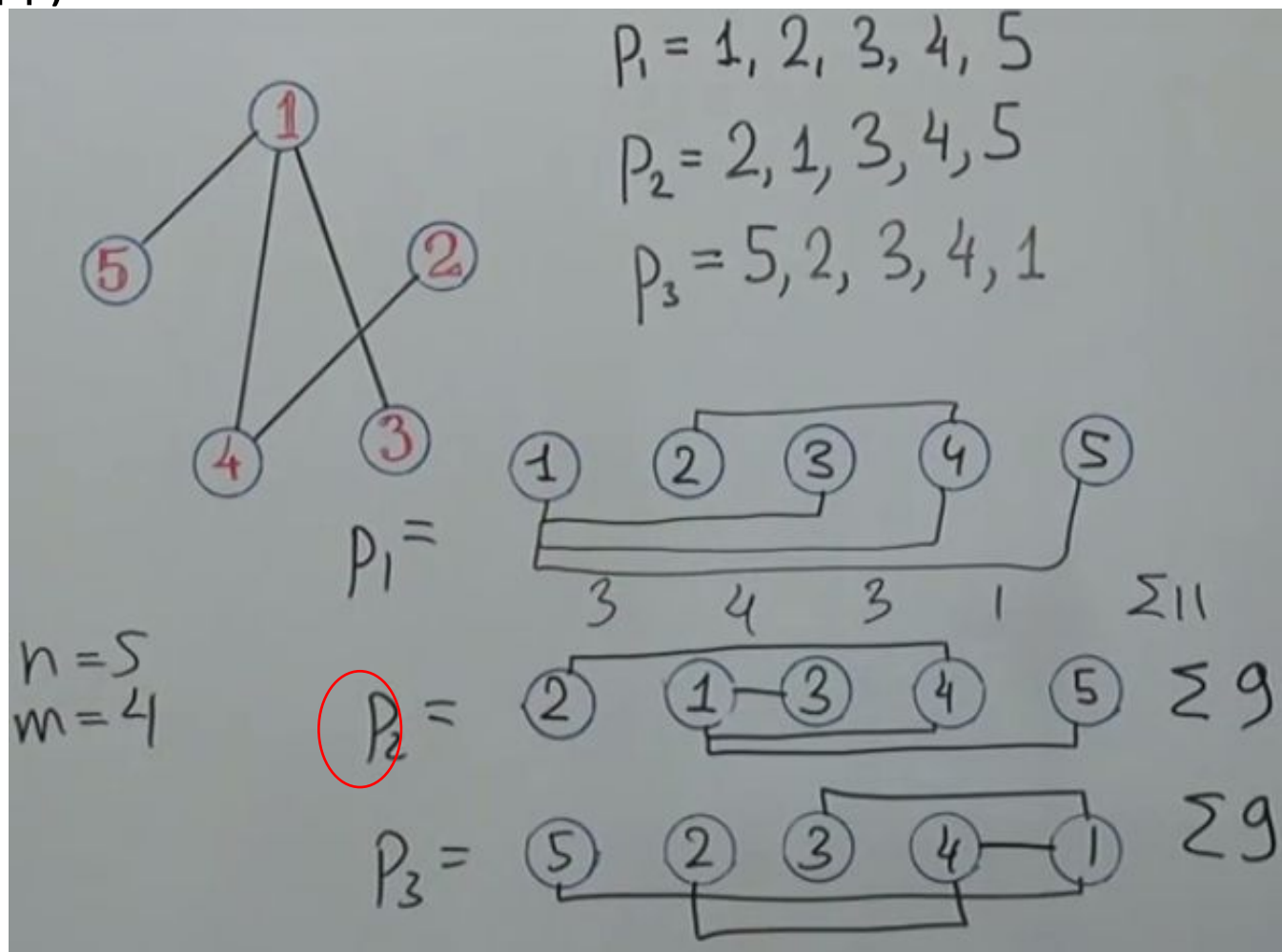
		c	b	a
x	$\boxed{-1}$	$\boxed{0}$	$\boxed{2}$	$\boxed{2}$
y	$\boxed{1}$	$\boxed{1}$	$\boxed{-2}$	$\boxed{-1}$
	-0,25	0	0,154	0,286

x	$\boxed{0}$	$\boxed{2}$	$\boxed{2}$	$\boxed{2}$
y	$\boxed{-1}$	$\boxed{-1}$	$\boxed{1}$	$\boxed{-2}$
	0	0,286	0,286	0,154

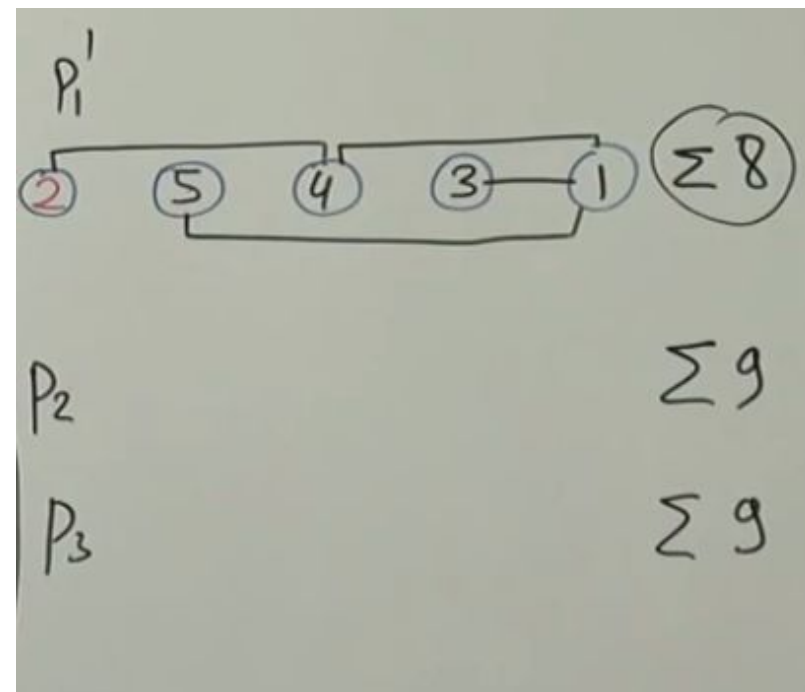
x	$\boxed{2}$	$\boxed{2}$	$\boxed{2}$	$\boxed{2}$
y	$\boxed{1}$	$\boxed{1}$	$\boxed{-2}$	$\boxed{-1}$
	0,286	0,286	0,154	0,286

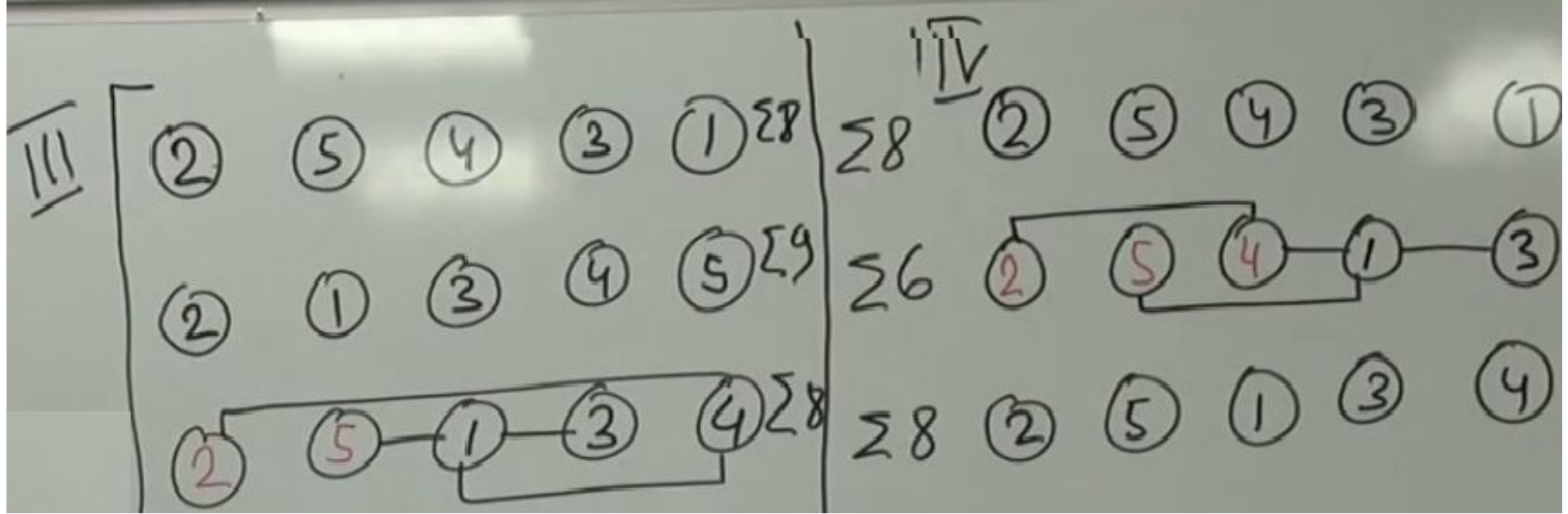
Размещение графа на линейки

- За счет одних мутаций – инверсии по первым элементам (по 1, по 2 и т. д.)



$a, b, c, d, \dots, z$ $a, z, y, x, \dots, b$

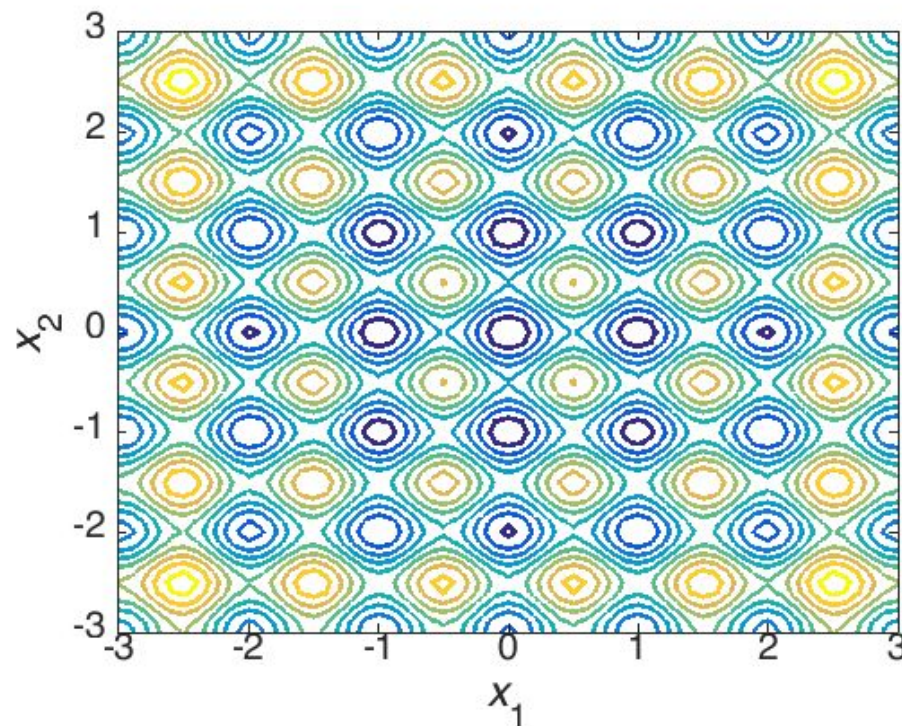
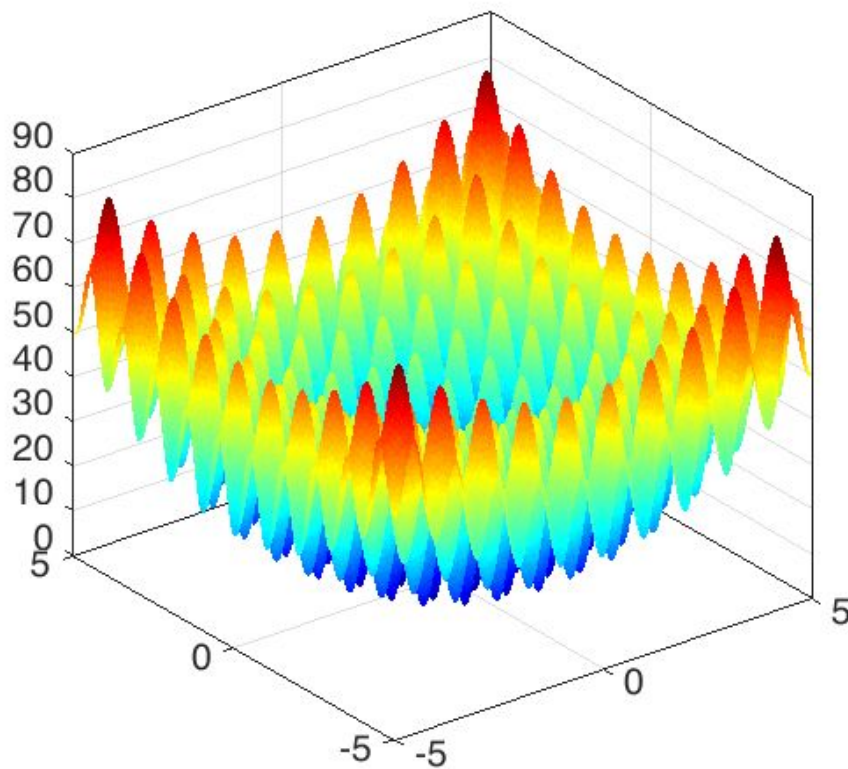




Пример: функция

$$f(x_1, x_2) = 20 + x_1^2 + x_2^2 - 10 \cos(2\pi x_1) - 10 \cos(2\pi x_2)$$

Имеет множество локальных минимумов и один глобальный
Метод Левенберга-Марквардта и ему подобные зачастую находят
локальный минимум



Пример: функция

Шаг 1. Создание начальной популяции

$$\begin{aligned}x_1^{(0)} &= x_2^{(0)} = 10 \\x_1^{(i)} &= x_1^{(0)} + N_i(0; \sigma^2) \\x_2^{(i)} &= x_2^{(0)} + N_i(0; \sigma^2)\end{aligned}$$

Шаг 2. Отбор

Целевая функция – значение функции Растригина для $x_1^{(i)}$ и $x_2^{(i)}$

Шаг 3. Скрещивание и мутации

Ведётся на основе лучших 5-10% членов популяции

Скрещивание:

$$\begin{cases} x_1^{(i)} = (1 - w_1)x_1^{(m)} + w_1x_1^{(n)}; w_1 \in [0; 1] \\ x_2^{(i)} = (1 - w_1)x_2^{(m)} + w_2x_2^{(n)}; w_2 \in [0; 1] \end{cases}$$

Мутация:

$$\begin{cases} x_1^{(i),*} = x_1^{(i)} + N(0; \sigma^2) \\ x_2^{(i),*} = x_2^{(i)} + N(0; \sigma^2) \end{cases}$$

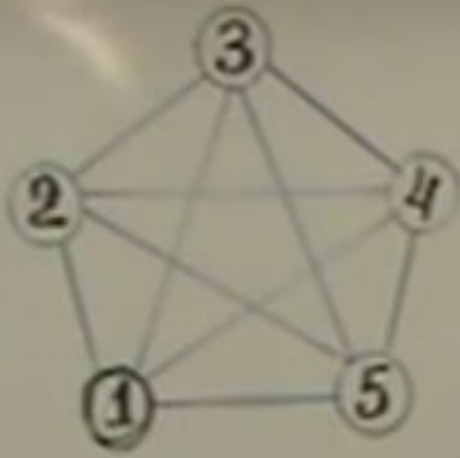
Муравьиный алгоритм

- **Муравьиный алгоритм** (*алгоритм оптимизации подражанием муравьиной колонии*, [англ. ant colony optimization, ACO](#)) — один из эффективных [полиномиальных алгоритмов](#) для нахождения приближённых решений [задачи коммивояжёра](#), а также решения аналогичных задач поиска маршрутов на [графах](#). Суть подхода заключается в анализе и использовании модели поведения [муравьёв](#), ищущих пути от колонии к источнику питания и представляет собой метаэвристическую оптимизацию. Первая версия алгоритма, предложенная доктором наук [Марко Дориго](#) в [1992 году](#), была направлена на поиск оптимального пути в графе.
- В основе алгоритма лежит поведение [муравьиной колонии](#) — маркировка более удачных путей большим количеством [феромона](#). Работа начинается с размещения муравьёв в вершинах графа (городах), затем начинается движение муравьёв — направление определяется вероятностным методом

Муравьиный алгоритм для задачи коммивояжера

τ - количество
феромона

$i-j$	L_{ij}	τ_{ij}
1-2	38	3
1-3	74	2
1-4	59	2
1-5	45	2
2-3	46	1
2-4	61	1
2-5	72	1
3-4	49	2
3-5	85	2
4-5	42	1



Hamilton W
1859

$$P_{ij} = 100 \frac{\eta_{ij}^{\beta} \tau_{ij}^{\alpha}}{\sum_k \eta_{ik}^{\beta} \tau_{ik}^{\alpha}}$$

$\alpha=1 \quad \beta=1$

Marco Dorigo

α - эк. ант.

$$\beta=0$$

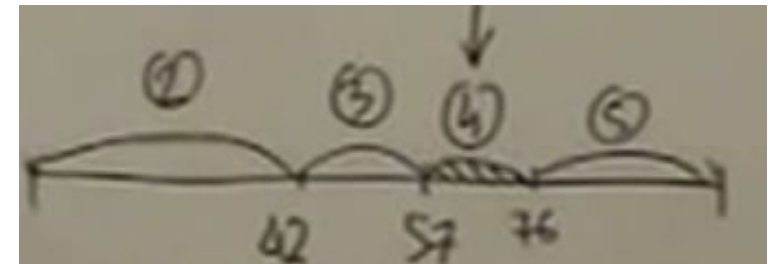
$$P_{ij} = \frac{1}{L_{ij}}$$

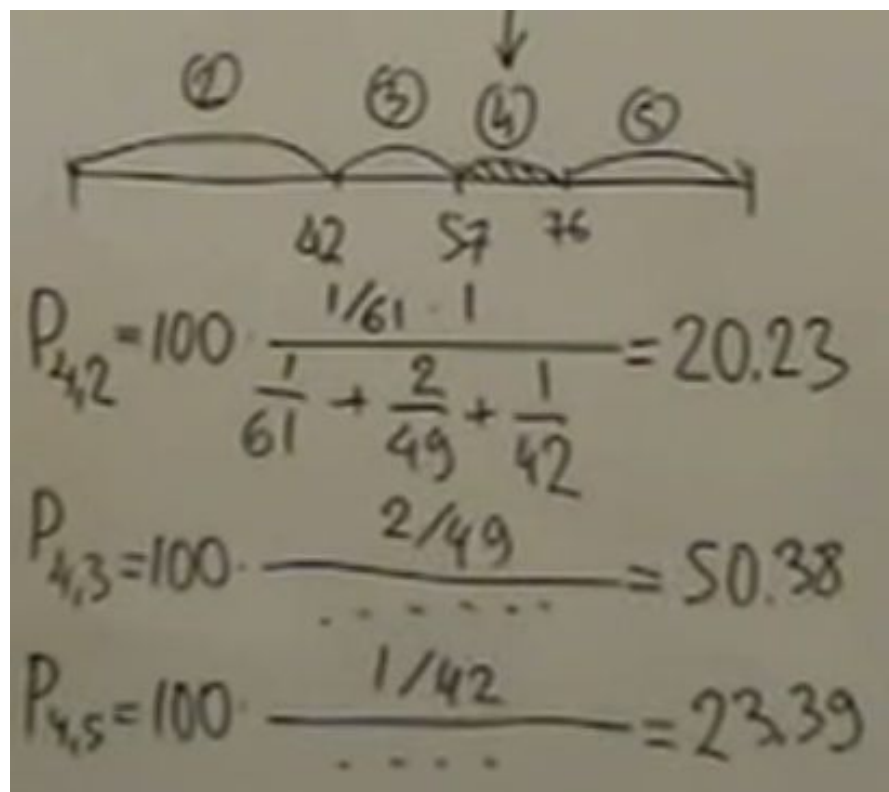
$$P_{32} = 100 \frac{\frac{1}{38} \cdot 3}{\frac{3}{38} + \frac{2}{74} + \frac{2}{59} + \frac{2}{45}} = \frac{7.9}{0.18} = 42.83$$

$$P_{43} = 100 \frac{\frac{1}{74} \cdot 2}{0.18} = 14.66$$

$$P_{14} = 100 \frac{\frac{1}{59} \cdot 2}{0.18} = 18.4$$

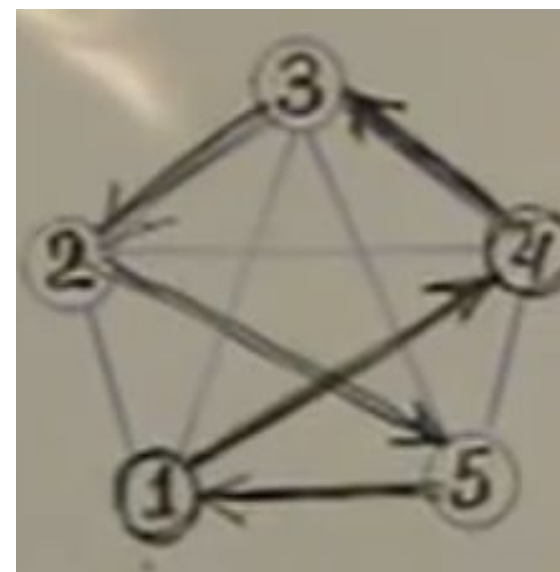
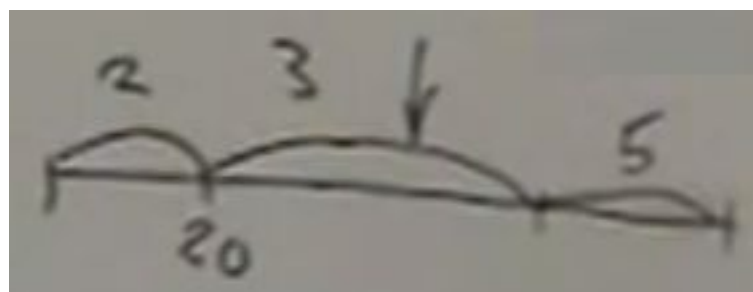
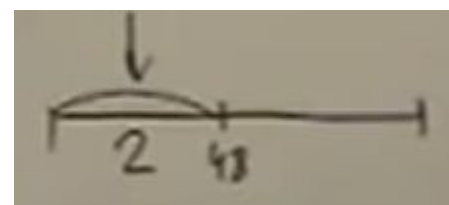
$$P_{15} = 100 \frac{\frac{1}{45} \cdot 2}{0.18} = 24.11$$





$$P_{3,2} = 100 \cdot \frac{\frac{1}{46} \cdot 1}{\frac{1}{46} + \frac{2}{85}} = 48.02$$

$$P_{3,5} = 100 \cdot \frac{2/85}{\dots} = 51.98$$



Корректируем феромоны

$$\sum_k L_k = L_0$$

$$\Delta\tau = \frac{Q}{L_0}$$
$$\tau_{kj}^* = \tau_{kj} + \Delta\tau$$
$$\tau_{kj}^* = (1-p)\tau_{kj} + \Delta\tau$$

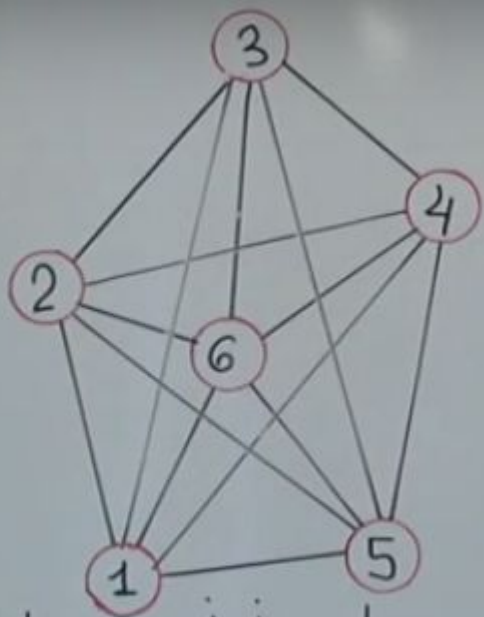
Q – константа

p –
забывчивость

Метод отжига

- **Алгоритм имитации отжига** ([англ. Simulated annealing](#)) — общий алгоритмический [метод](#) решения задачи глобальной оптимизации, особенно дискретной и [комбинаторной оптимизации](#). Один из примеров [методов Монте-Карло](#).
- Алгоритм основывается на [имитации физического процесса](#), который происходит при [кристаллизации вещества](#), в том числе при [отжиге металлов](#). Предполагается, что [атомы](#) уже выстроились в [кристаллическую решётку](#), но ещё допустимы переходы отдельных атомов из одной ячейки в другую. Предполагается, что процесс протекает при постепенно понижающейся [температуре](#). Переход атома из одной ячейки в другую происходит с некоторой [вероятностью](#), причём вероятность уменьшается с понижением температуры.

Метод отжига

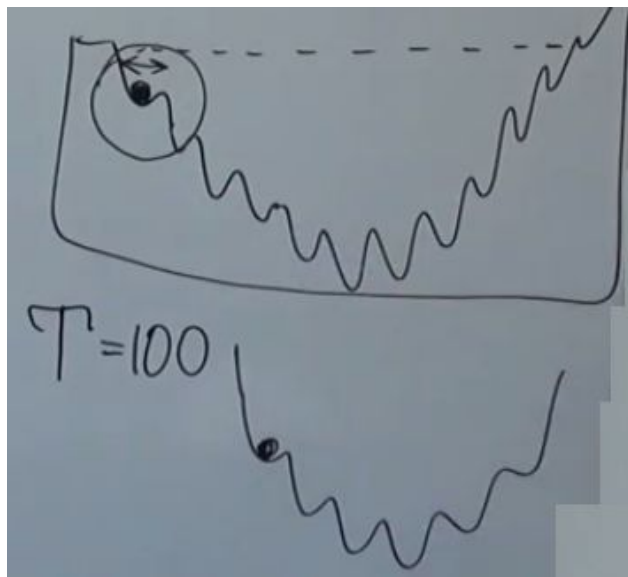


i, j	L_{ij}	i, j	L_{ij}
1-2	19	2-6	13
1-3	41	3-4	20
1-4	39	3-5	41
1-5	27	3-6	22
1-6	20	4-5	26
2-3	24	4-6	20
2-4	31	5-6	23
2-5	35		

$$S = [1, 4, 3, \overbrace{5, 6}, 2, 1]; L = 155$$

$$S' = \overset{4,6}{[1, 4, 3, 2, \overbrace{6, 5}, 1]} \quad L' = 146$$

$$S'' = \overset{5,6}{[1, 4, 3, 2, \overbrace{5, 6}, 1]} \quad L'' = 161$$



$$p^* = 100 e^{-\frac{\Delta S}{T}} = 100 e^{-\frac{13}{50}} = (74,62) > [62]$$

$$\Delta S < 0 \quad p^* > p_k$$

$$T_{k+1} = \alpha T_k$$

T-
температура

$$(3,5) [1, 4, 5, 2, 3, 6, 1] \quad L''' = 166$$

$$P^* = 100 e^{-\frac{5}{25}} = 81,87 < 83$$

$$(6,2)$$

$$S = [1, 6, 3, 2, 5, 4, 1]$$

$$L = 166$$

$$P^* = 100 e^{-\frac{5}{12,5}}$$