

Конспект лекций

Введение в эволюционные вычисления

Акира Имада

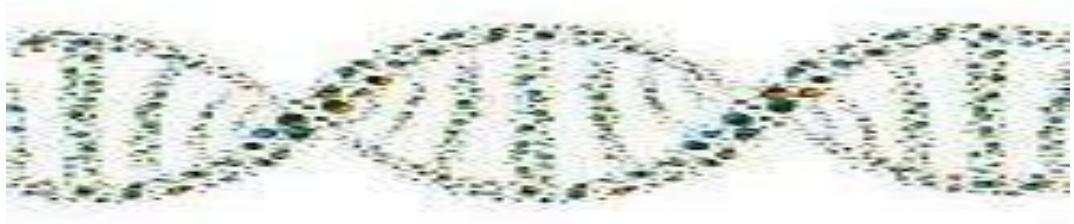
Брестский государственный технический университет,
Беларусь

Перевел Бегеза К.С.

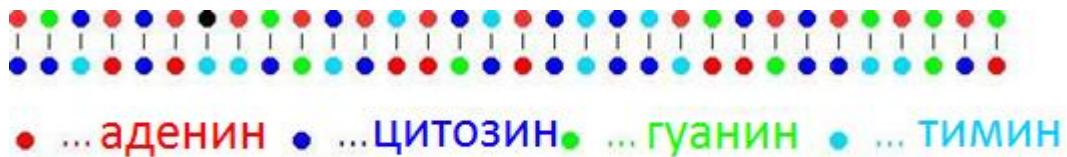
Дата последнего изменения **15 мая 2015**

1. Идея заимствована из биологической эволюции

Изображение спирали ДНК



Упрощение

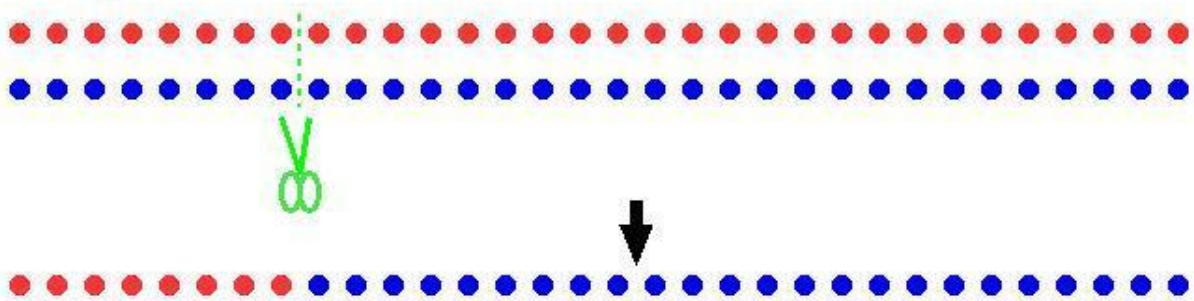


Наша виртуальная хромосома

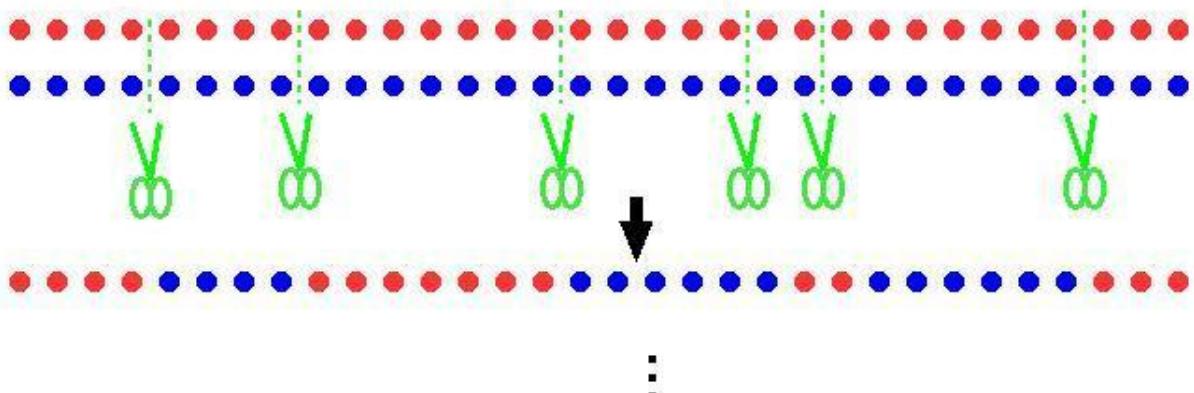


Кроссовер

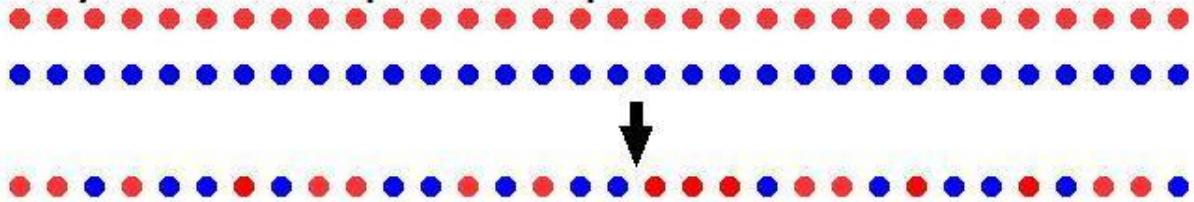
Одноточечный Кроссовер



Многоточечный Кроссовер

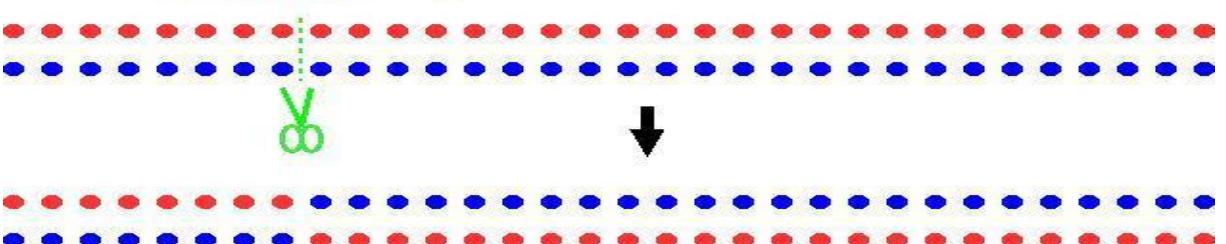


Случайный Кроссовер



Одноточечный кроссовер по случайному номеру

If `rand = 8`



Единый(случайный) кроссовер по случайному числу

Мутация

$$\text{вероятность мутации} = \frac{1}{\text{количество генов}}$$

Создаем случайное число от 0 до N-1 каждому из всего количества генов

Если и только если случайное число = 0 тогда мутация



Урезанная выборка

Выбираем родителей

Создаем 2 случайных числа от 1 до N/2

```
graph TD; A[Кроссовер  
создание ребенка  
для следующего  
поколения] --> B[Повторить N раз]; A --> C[Урезать!];
```

Кроссовер
создание ребенка
для следующего
поколения

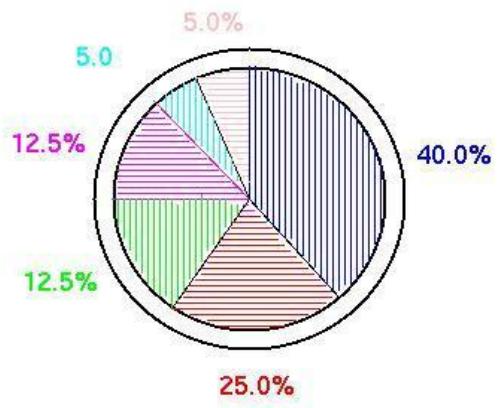
Повторить **N** раз

>8 Урезать!

Приспособленность (Fitness) Пропорциональной выборки

fitness

#1	16	$=> 16/40 = 0.400$
#2	10	$=> 10/40 = 0.250$
#3	5	$=> 5/40 = 0.125$
#4	5	$=> 5/40 = 0.125$
#5	2	$=> 2/40 = 0.500$
#6	2	$=> 2/40 = 0.500$
<hr/>		40



a.k.a

Отбор рулеткой



2. Простейший вариант – Хилл Климбинг

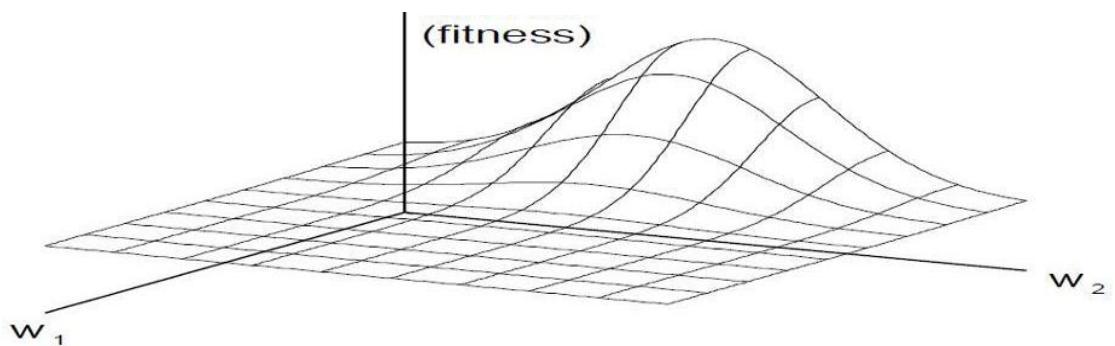
Случайная мутация Хилл-Климбинг

- (1) выбираем строку наугад и называем ее current-hilltop
- (2) выбрать локус наугад i_p . Если i_p приводит к равной или более высокой приспособленности затем установите current-hilltop к результирующей строке
- (3) шаг перейти (2) до оптимального строки был найден, или пока не будут выполнены макси-мума количество оценок.
- (4) возвращает текущее current-hilltop

3. 1-й пример. Все одна проблема

5. Пейзаж приспособленности(Fitness)

Концептуальный график значения приспособленности на 2-Д пространстве

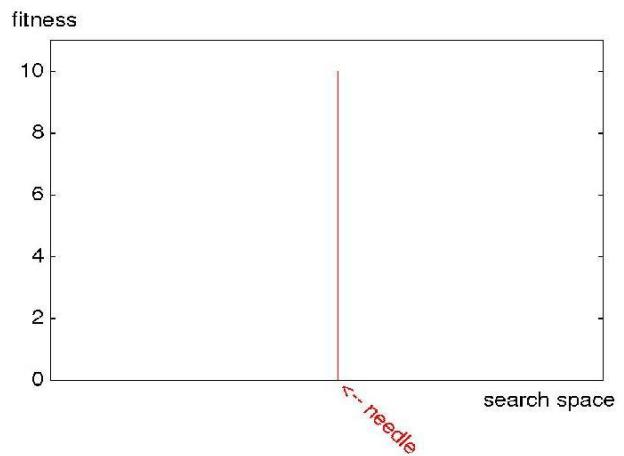


5. 2 пример. Проблема иглы в стоге сена

Проблема согласования N-значного пароля

Предположим, N-значного целого пароль и хромосомы с N целых генов, Фитнес 1, если и только если хромосома совпадает с паролем прекрасно, иначе фитнес = 0.

График приспособленности (Fitness) в этом случае

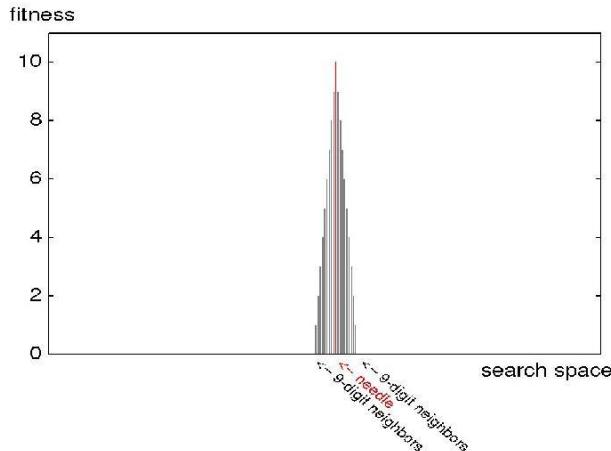


fitness	generation
0	5
1	986
2	3430
3	10152
4	25365
5	52982
6	95682
7	198253
8	402362
9	623023

(работа Бакуна Антона в 2015)

Расширенная оценка приспособленности

Когда мы оцениваем это по приспособленности «Сколько цифр совпадает?»



password	iteration	n
807	1473	3
3859	25237	4
03186	41489	5
723186	772246	6
9224186	3194682	7
35284196	16185762	8
874284695	78326820	9
7803204905	eror ! Very long time	10

(работа Бакуна Антона в 2015)

6. 3-й пример. Счастливая собака

Собака ищет сосиски

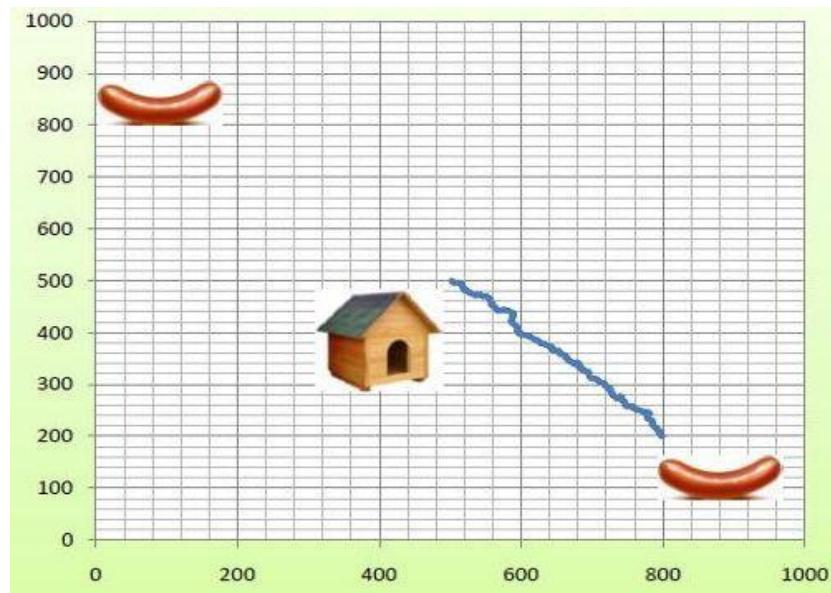
Собака находится в мире(0,0)-(1000,1000) с сосиской в (200,800)

Собака стартует в (500,500) .

7. Что делать, если существует более чем 1 решение?

Собака видит 2 сосиски

Собака в пространстве (0.0)-(1000,1000) с сосисками в (200, 800) и (800, 200). Собака стартует из (500, 500)



(Работа Белоус Софьи 2015)

По факту

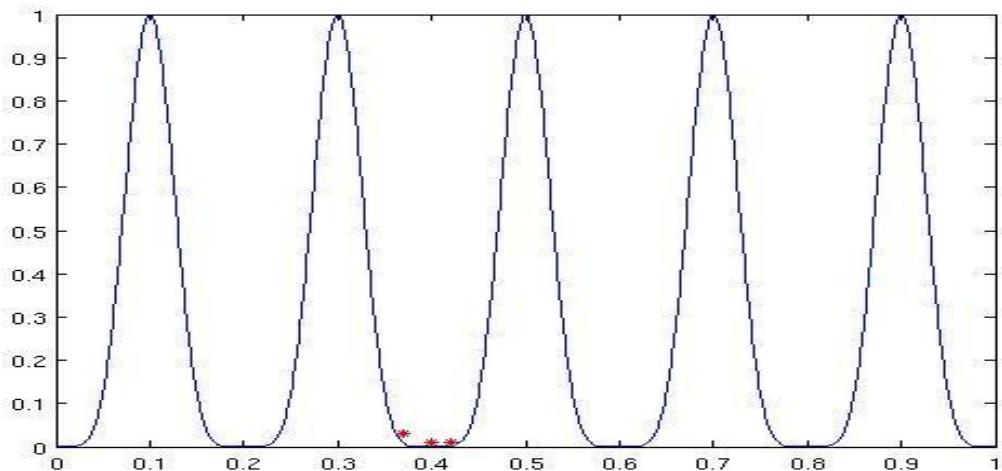
Все собаки собирают только один колбасу из двух!

↙

Можем ли мы осуществить таким образом, чтобы
половина собак перейти к одной, а другая половина в другой?

Минимизация 2-Д функции

$$y = \sin^6(5\pi x)$$



Два алгоритмы, чтобы получить все решения

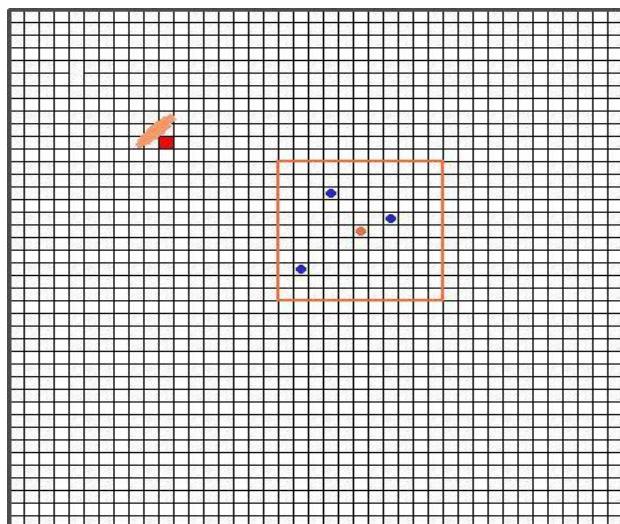
1. Фитнесс Шеринга

$$F_s(i) = \frac{F(i)}{\sum_{j=1}^{\mu} s(d_{ij})}$$

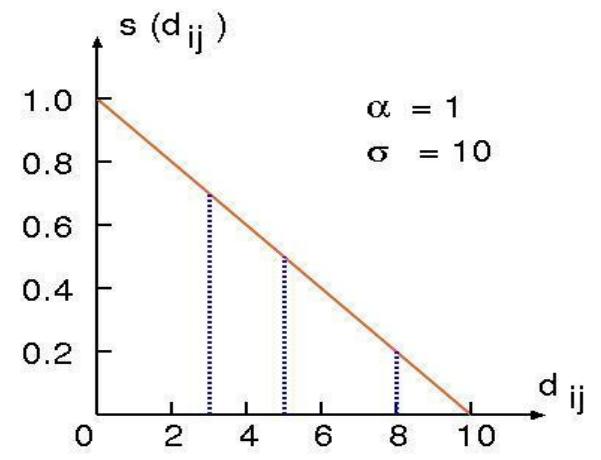
где

$$s(d_{ij}) = \begin{cases} 1 - (d_{ij}/\sigma_{\text{share}})^{\alpha} & \text{if } d_{ij} < \sigma_{\text{share}} \\ 0 & \text{otherwise} \end{cases}$$

Fitness Share - Lucky Dog



- ... sausage
- ... dog _i
- ... dog _j

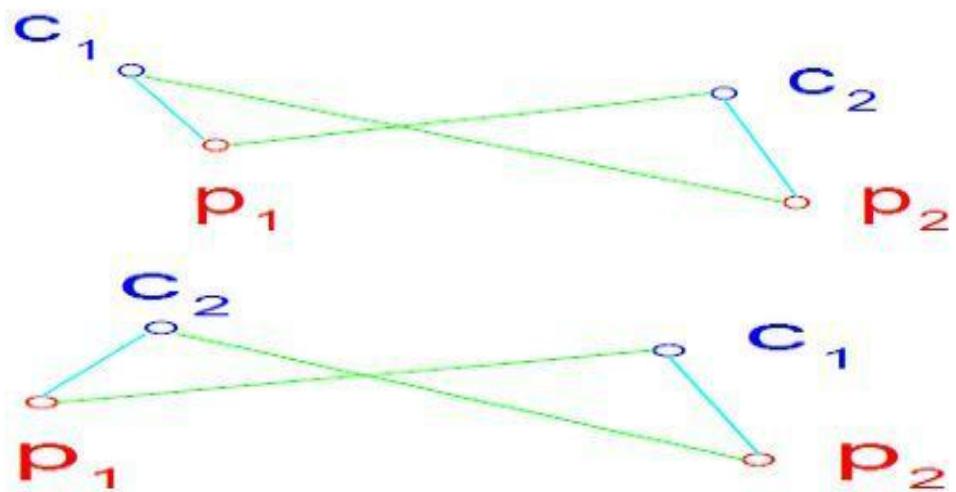


$$F_i = \frac{1600 - 20}{0.7 + 0.5 + 0.2}$$

2. Алгоритм вытеснения

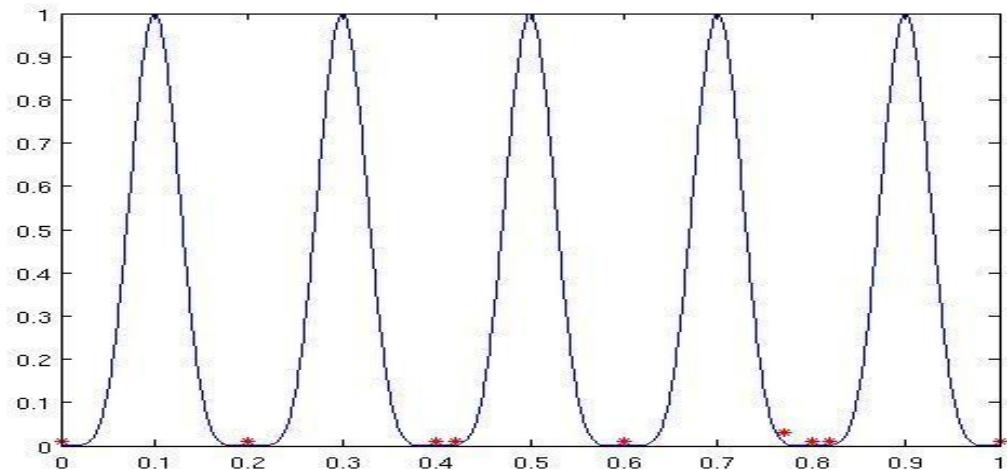
1. Выберите два родителей, $p1$ и $p2$, наугад.
2. Произведите двое детей, $c'1$ и $c'2$.
3. мутировать детей уступая $C1$ и $C2$, с кроссовером.
4. Замените родителя с ребенком следующим образом:
 - Если $D(p1, C1) + D(p2, c2) > d(p1, C2) + D(p2, c1)$
 - * Если $F(C1) > F(p1)$ затем заменить $p1$ с $c1$
 - * Если $F(c2) > F(p2)$ и замените $P2$ с $c2$
 - ЕЩЕ
 - * Если $F(c2) > F(p1)$ затем заменить $p1$ с $c2$
 - * Если $F(C1) > F(p2)$ и замените $P2$ с $c1$

Давка - два случая родителей и детей



2-D минимизация функции - Что происходит?

$$y = \sin^6(5\pi x)$$



8. Обычно используется функция тестирования

1. Сферическая модель

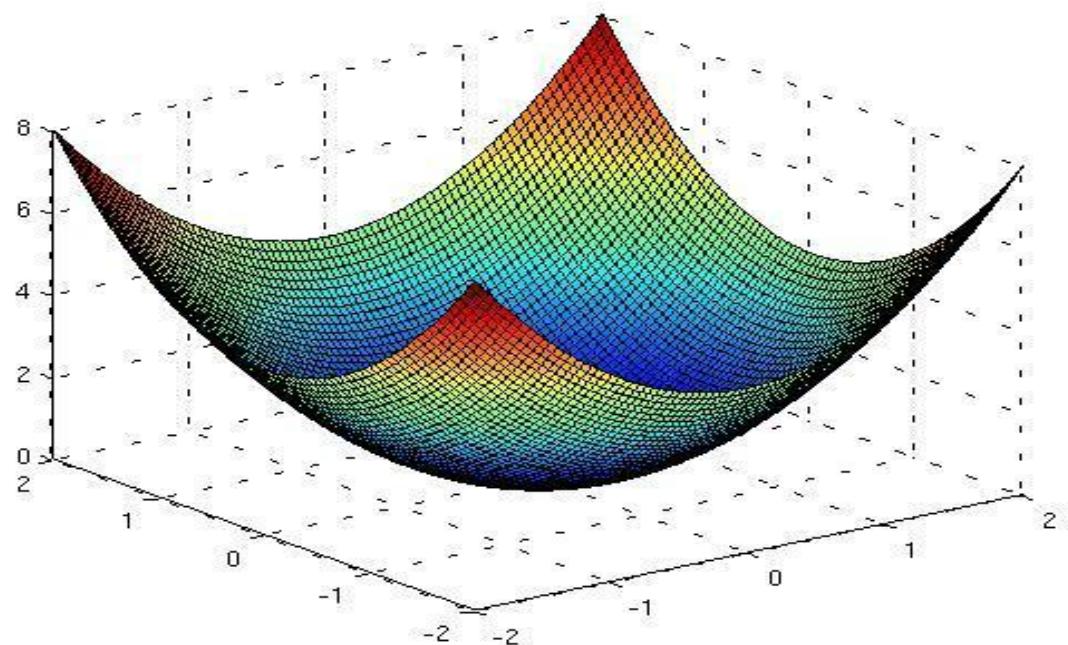
$$y = \sum_{i=1}^{20} x_i^2$$

то есть,

$$y = x_1^2 + x_2^2 + x_3^2 + \dots + x_{20}^2$$

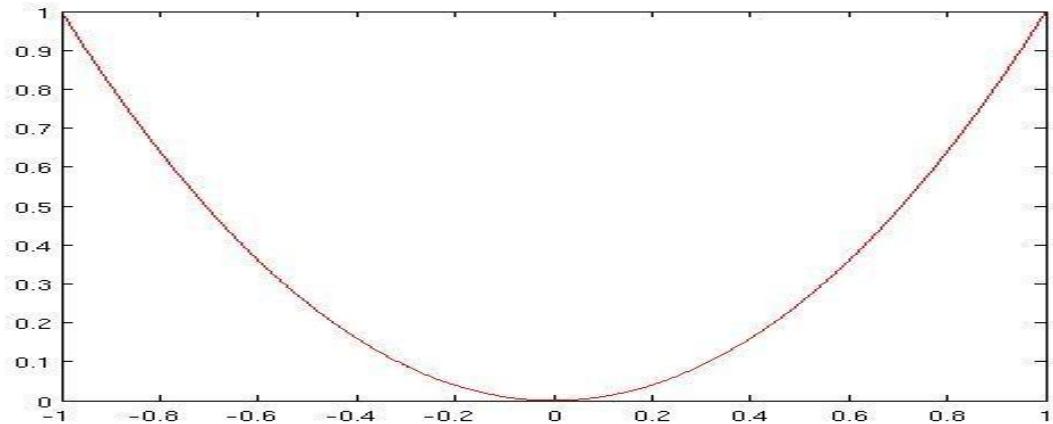
Это 3-Д версия

$$z = x^2 + y^2$$



Это 2-Д версия

$$y = x^2$$



2.Функция Швефеля

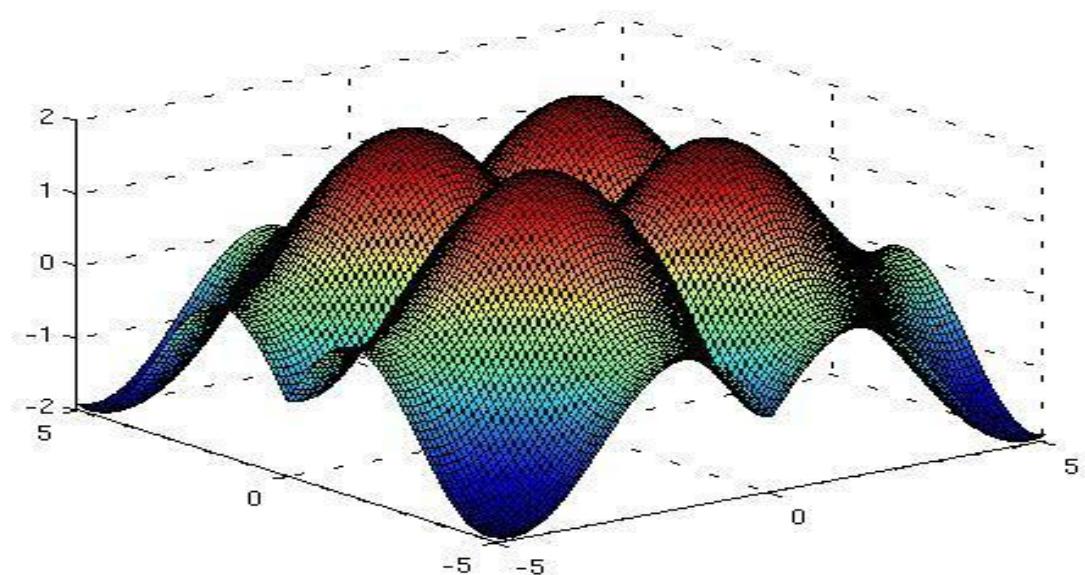
$$y = \sum_{i=1}^n (x_i \sin(|x_i|))$$

То есть,

$$y = x_1 \sin(|x_1|) + x_2 \sin(|x_2|) + \dots + x_{20} \sin(|x_{20}|)$$

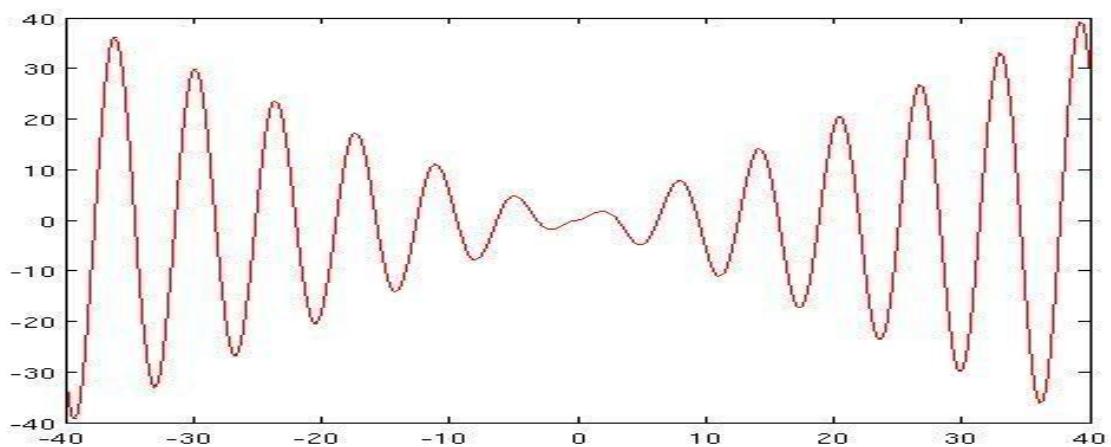
Швефель – 3д версия

$$z = x \sin(|x|) + y \sin(|y|)$$



Швефель – 2-д версия

$$y = x \sin(|x|)$$

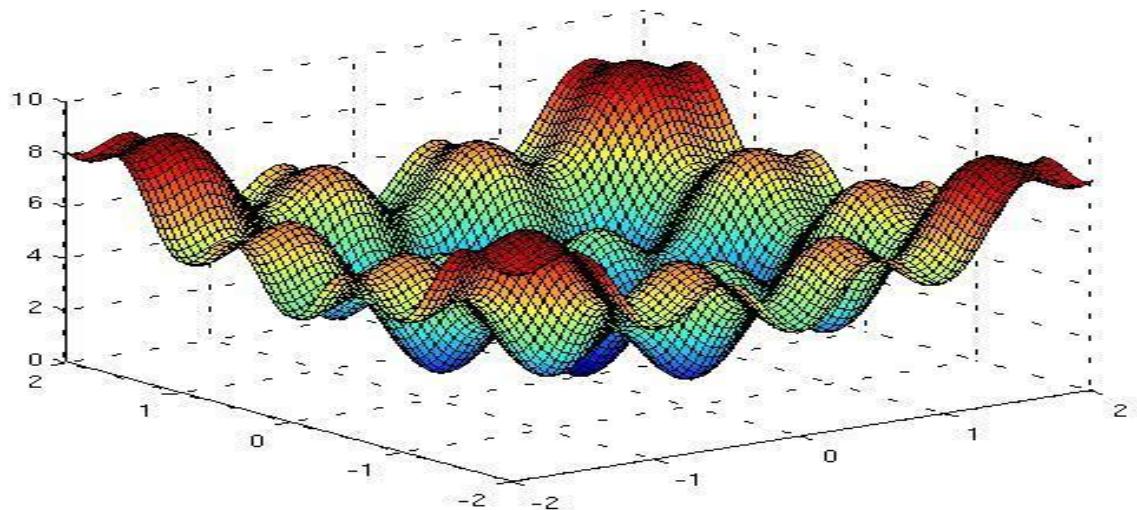


3. Функция Растринга

$$y = nA + \sum_{i=1}^n (x_i^2 - A \cos(2\pi x_i))$$

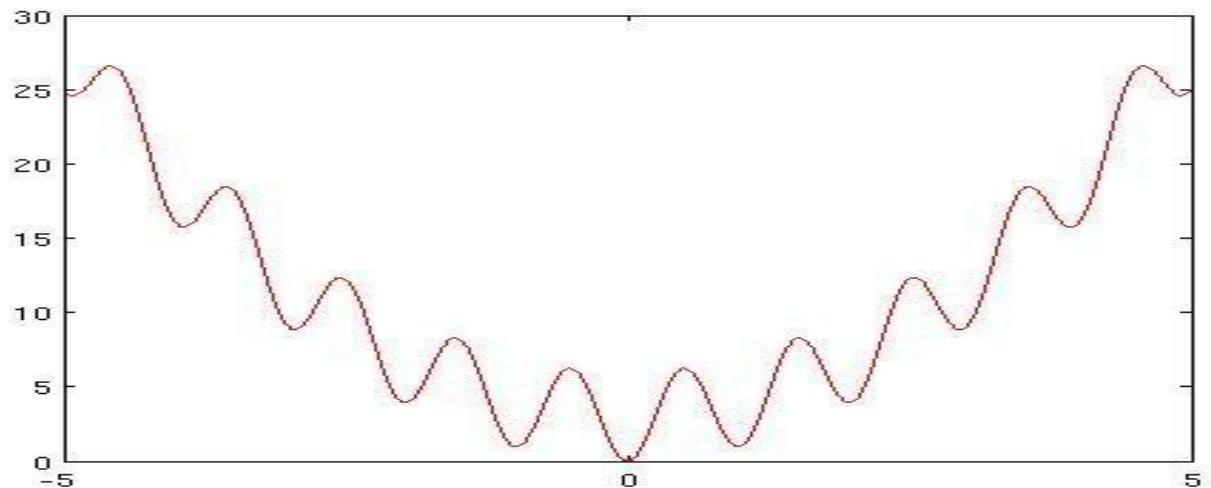
Растринг – 3-д версия

$$z = 2 + x^2 - \cos(2\pi x) + y^2 - \cos(2\pi y)$$



Растринг – 2-д версия

$$y = 3 + x^2 - 3 \cos(2\pi x)$$

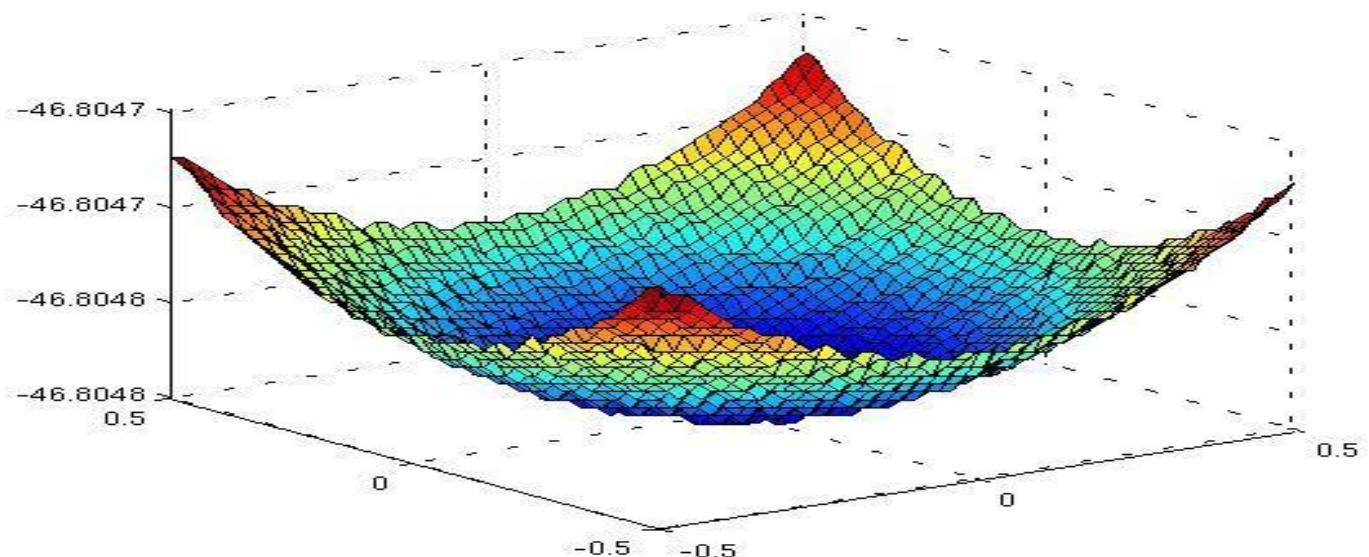


4. Функция Griewangk

$$y = \sum_{i=1}^n x_i^2 / 4000 - \prod_{i=1}^n \cos(x_i / \sqrt{i}) + 1$$

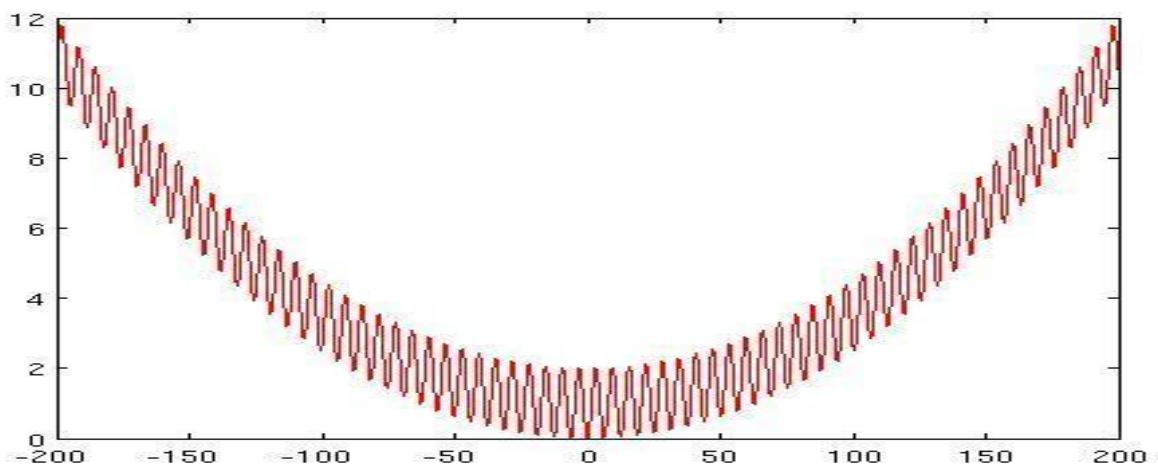
Griewangk – 3-д версия

$$z = \frac{x^2 + y^2}{4000} - (\cos x) \cos(x/\sqrt{2}) + 1$$



Griewangk – 2-д версия

$$y = \frac{x^2}{4000} - \cos x + 1$$

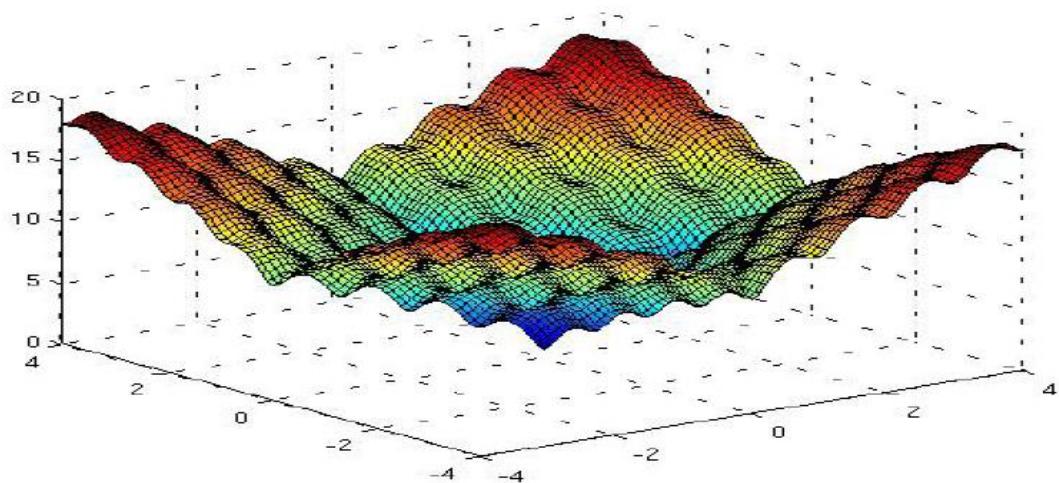


5. Функция Экли

$$y = -20 \sum_{i=1}^n \exp(-0.2 \sqrt{x_i^2/n}) - \exp((\sum_{i=1}^n \cos 2\pi x_i)/n) + 20 + e$$

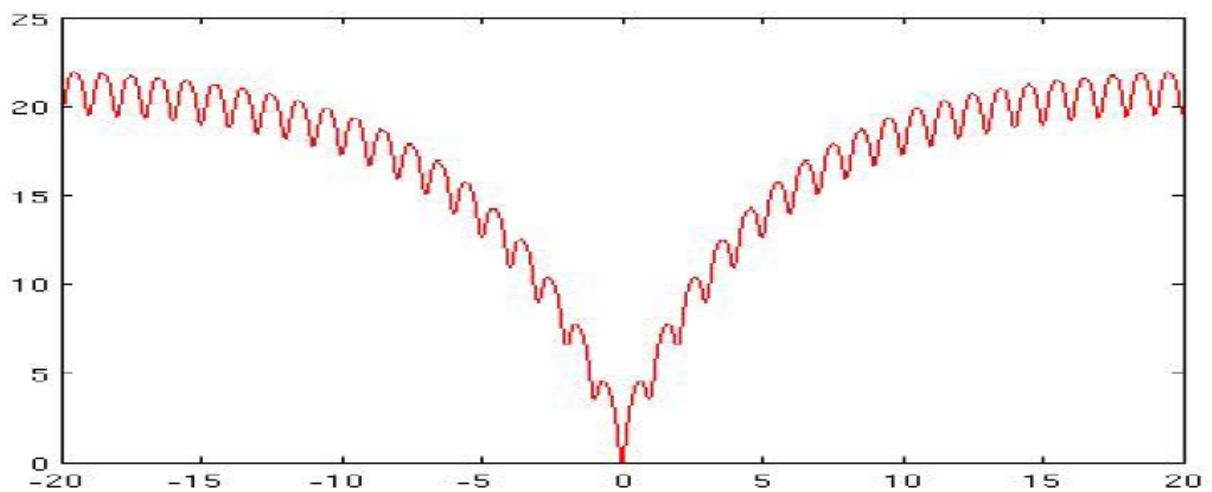
Функция Экли – 3-д версия

$$z = -20 \exp(-0.2 \sqrt{(x^2 + y^2)/2}) - \exp(\cos 2\pi x + \cos 2\pi y)/2) + 20 + e$$



Функция Экли – 2-д версия

$$y = -20 \exp(-0.2\sqrt{x^2}) - \exp(\cos 2\pi x) + 20 + e$$



9.Что делать, если больше, чем 1 функция фитнесс?

Оптимальное решение Parete

Если X лучше, чем Y для всех приспособленности, то говорят, что **X доминирует Y** .

Когда раствор не доминирует любые другие его называют **не преобладающее решение**

или

Оптимальное решение Parete

Ранг - Когда мы пытаемся эволюции для оптимального Parete

Заказать в соответствии с
количеством, как и многие другие в популяции он
доминирует

Например. кто доминирует, кому и как ранг засчитывается.

	test-1	test-2	test-3	test-4	test-5	dominated by	dominates	rank
A	9	9	9	8	7	0	3	1
B	5	4	5	3	6	1	2	2
C	3	3	4	2	3	2	1	3
D	2	2	3	1	2	3	0	4
E	1	1	1	1	9	0	0	4

Алгоритм

- 1. Инициализировать население.*
- 2. Выберите индивиды равномерно из населения.*
- 3. Выполните кроссовера и мутации для создания ребенка.*
- 4. Рассчитать звание нового ребенка.*
- 5. Найдите человека во всей популяции, наиболее похож на ребенка. Замените это лицо с новым ребенком, если рейтинг ребенка лучше, или если ребенок доминирует его.*
- 6. Обновление рейтинга населения, если ребенок был вставлен.*
- 7. Выполните шаги 2-6 в зависимости от размера населения.*
- 8. Если критерий остановки не выполняется переход к шагу 2 и начать новое поколение.*

Упражнение

Попробуйте его с двумя функциями приспособленности $Y1 = (x - 2)^2$ и $y2 = (x - 4)^2$ следующим образом:

1. Создать 20 10-битный двоичный хромосомы, предполагая, каждая хромосома представляют

x-coordinate колеблется от 0 до 6 с (0000 • • • 00) и (1111 • • • 11) будучи соответствующие 0 и 6, соответственно.

2. Рассчитать $y1$ и $y2$ для каждого из 20 x годов, представленных этими 20 хромосом.

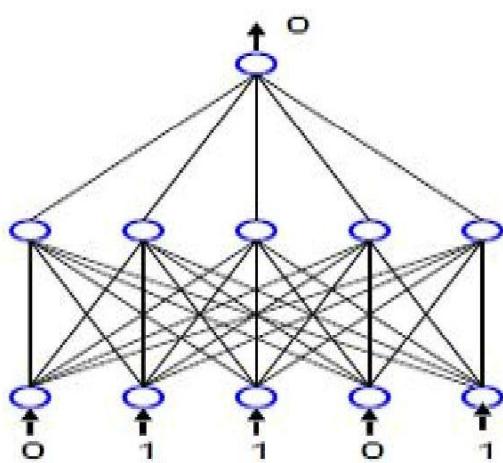
3. Создать таблицу с 5 столбцов: (я) хромосома, (II) его значение x , (III) его значение $y1$, (IV) его значение $y2$, (v), сколько эти ($Y1$, $Y2$) доминирует других (ранг).

4. Участок эти 20 очков на каждом из двух графиков (например, красный и синий цвета).

5. Создайте новое поколение путем применения алгоритма на этих 20 хромосом.

6. В то время как 20 пунктов различны от предыдущего поколения У 2-4 иначе остановку.

10. Вес конфигурационной эволюции нейронной сети



input	output
00000	1
00001	0
00010	0
00011	1
00100	0
00101	1
00110	1
00111	0
01000	0
...	...
11110	1
11111	0

Нейронная сеть с прямой связью

На выходе y с исп. 1 нейрона с 5 входами $+1$ or -1 это рассчитывается:

$$y = \operatorname{sgn}\left(\sum_{i=1}^5 w_i x_i\right)$$

если $\sum_{i=1}^5 w_i x_i$ является отрицательным, то $y = -1$, в противном случае $y = 1$

Эволюция веса конфигурационном

Хромосома:

$(w_1, w_2, w_3, \dots, w_{30})$

Фитнесс:

сколько выходов нужно собрать

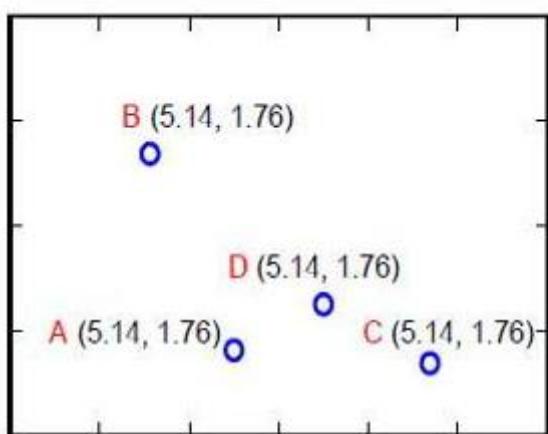
когда каждый из всех возможных $2^5 = 32$ входов дается

один за другим?

10.Задача путешествующего продавца(TSP)

TSP с 4 городами

карта 4 городов



матрица расстояний

	A	B	C	D
A	0.00	4.13	4.34	1.95
B	4.13	0.00	7.35	5.00
C	4.34	7.35	0.00	2.51
D	1.95	5.00	2.51	0.00

Все возможные пути

$$A-B-C-D-A \Rightarrow 4.13 + 7.35 + 2.51 + 1.95 = 15.94$$

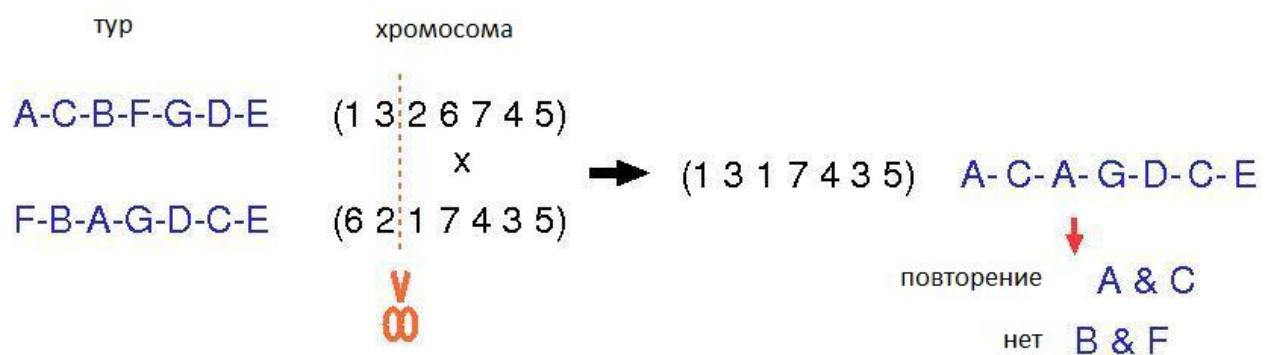
$$A-B-D-C-A \Rightarrow 4.13 + 5.00 + 2.51 + 4.34 = 15.98$$

$$A-B-C-D-A \Rightarrow 4.34 + 7.35 + 5.00 + 1.95 = 18.64$$

Например. Путешествие (А-С-В-Ж-С-Д-Е).

Может ли он быть закодированы (1 3 2 6 3 4 5)? ↓ Ответ: нет!

Одноточечный кроссовер, не работает



Тогда как для кодирования TSP в хромосому?

Шаг 1. Установите $i = 1$.

Шаг 2. Если i -тый ген n , то n -й город в списке является городом, в настоящее время посещения.

Шаг 3. Удалить город из списка.

Шаг 4. Установите $i = i + 1$ и повторите шаг-2 Шаг-4, пока $i \leq n$.

12. Итерированная дилемма заключенного.

Игра для двух игроков как бумага камень ножницы

Награды:

когда		каждый получит	
A	B	A	B
1	1	1	1
1	0	5	0
0	1	0	5
0	0	3	3

Повторная игра

Игра

	01	02	03	04	05	06	07	08	09	10	11	12	Σ
A													
B													

Попробуй: Случайный пр. Случайный и Лучший пр. Случайный, Всегда-1, Всегда-0

Хромосома как стратегия игры

Хромосома

A 1001000101011101110100101010010101111000010101100010101101010001
B 1101000101101110000101001111010100111101011010110101101010111011

Итерационная игра

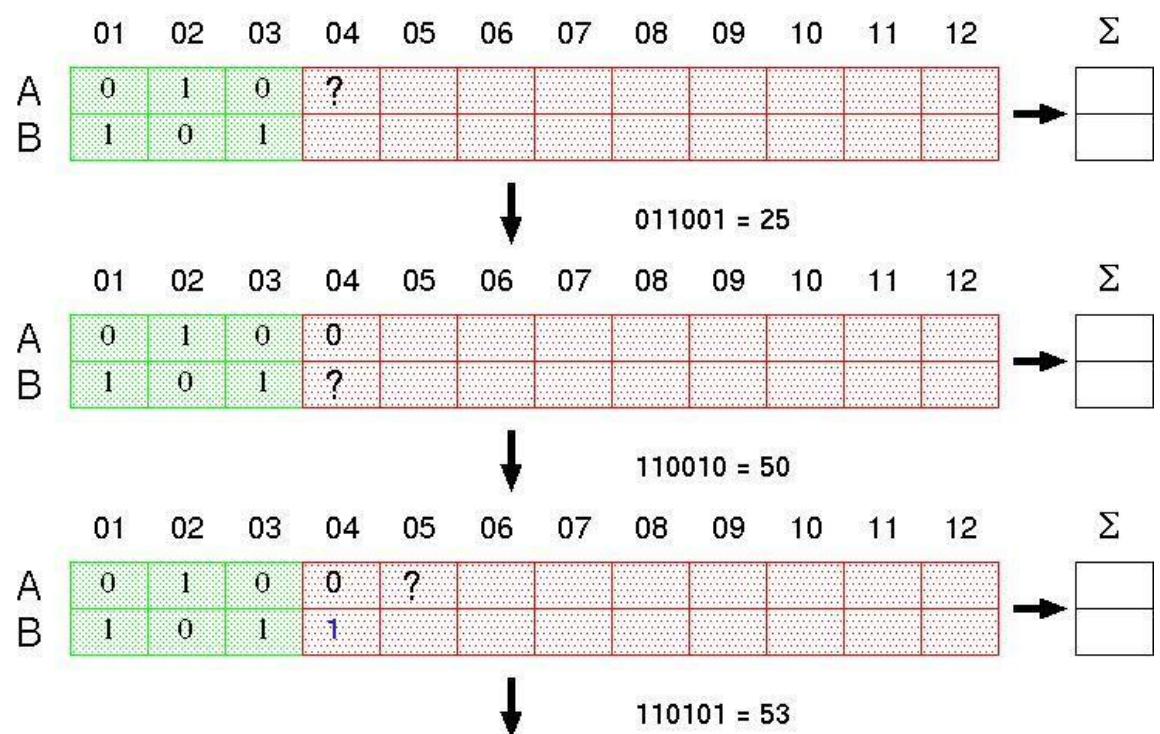
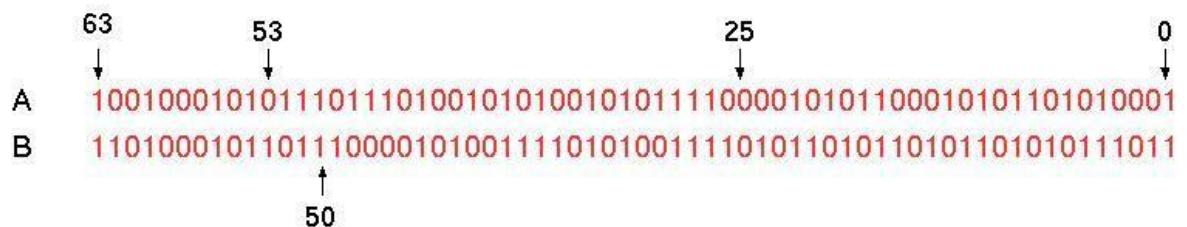
A: $010\boxed{?} \dots 0000$

B: $101 \quad 011\boxed{?}$

011001 = 25

01010 = 10

Дальнейшие действия зависят от хромосомы



13. Снижение измерения эволюции

Отображение Samon

3 точки в 3Д	Матрица расстояний				Нормализация			
	A	B	C	O	A	B	C	O
A: (0.328, 0.819, 0.118)	A 0.000	1.245	1.481	0.890	A 0.000	0.846	1.000	0.601
B: (0.129, -0.256, 0.713)	B	0.000	1.126	0.769	B	0.000	0.760	0.519
C: (0.277, -0.584, -0.354)	C		0.000	0.737	C		0.000	0.498
O: (0.000, 0.000, 0.000)	O			0.000	O			0.000

случайные 5 точек в 2Д

	X	Y	Z	O	X	Y	Z	O
X: (0.514, -0.223)	X 0.000	1.826	0.632	0.560	X 0.000	1.000	0.346	0.307
Y: (-0.861, 0.979)	Y	0.000	1.348	1.304	Y	0.000	0.738	0.714
Z: (-0.113, -0.142)	Z		0.000	0.182	Z		0.000	0.273
O: (0.000, 0.000)	O			0.000	O			0.000

Хромосома

(0.514 -0.223 -0.861 0.979 -0.113 -0.142 0.000 0.000)

Приспособленность

$$(0.846-1.000)+(1.000-0.346)+(0.601-0.307)+(0.760-0.738)+(0.519-0.714)+(0.498-0.273)$$

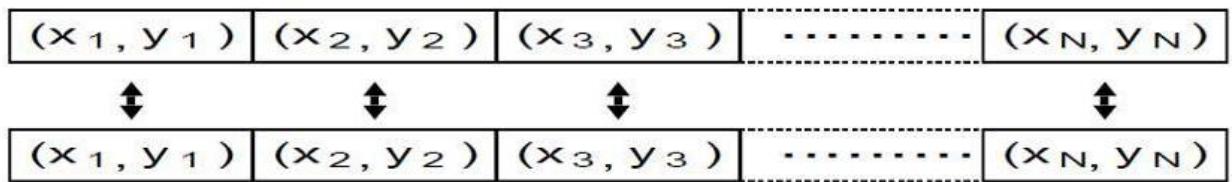
$$= 0.846$$

Хромосома для отображения Samon

Хромосома

(x_1, y_1)	(x_2, y_2)	(x_3, y_3)	(x_N, y_N)
--------------	--------------	--------------	-------	--------------

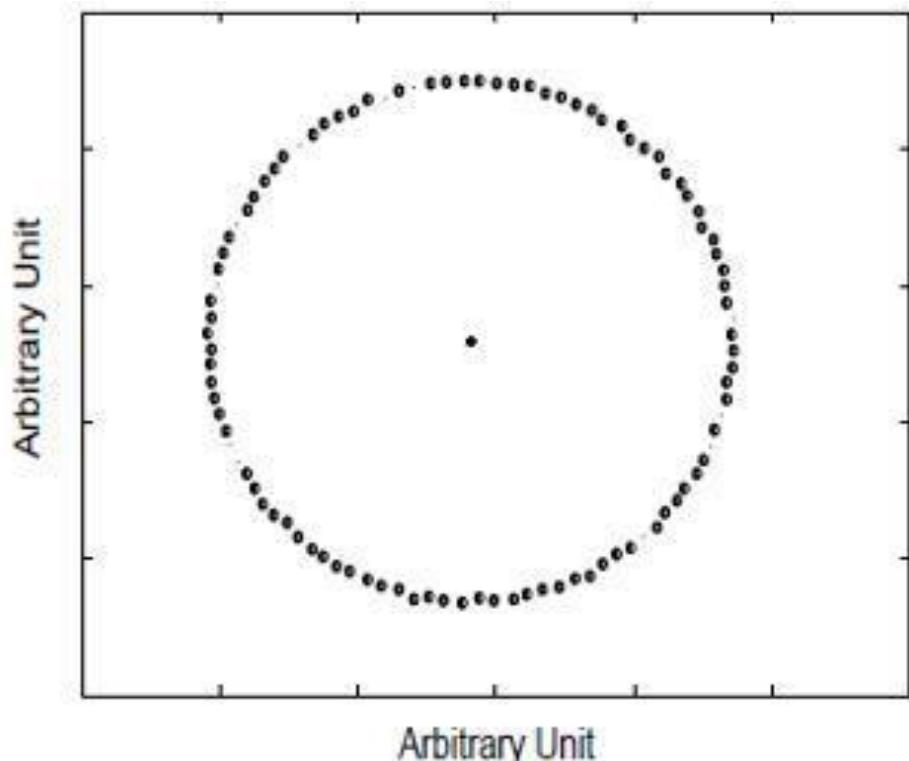
Рекомбинация с единым кроссовером



Алгоритм

- 1. Пусть N точек приведены в n -D пространстве.*
- 2. Рассчитать матрицу расстояний R ($N \times N$), чья i - j элементом является евклидово расстояние между i -й и j -й точки.*
- 3. Также думать о предварительных N точек в 2-D пространстве, расположенных случайным образом в начале.*
- 4. Расстояние матрица Q рассчитывается таким же образом, как R .*
- 5. Тогда матрица ошибок $P = R - Q$ определяется.*
- 6. Поиск на местах N точек в 2-D пространстве, который минимизирует сумму элементов P .*

Пример отображает круг с поверхности гиперсферы

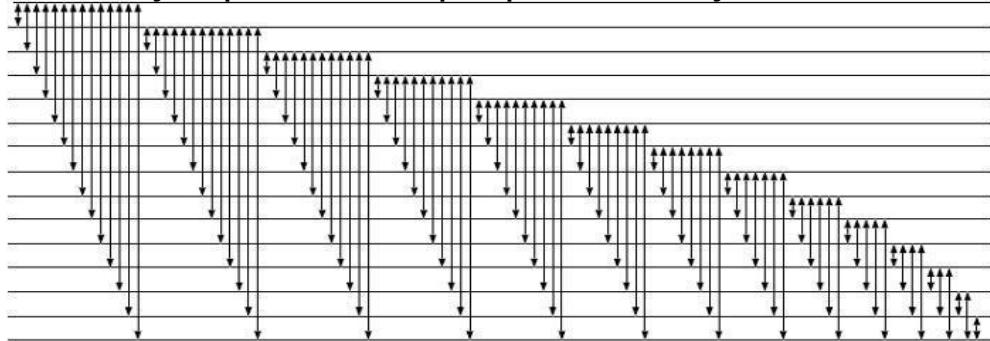


14. Сортировка сети

1.

2. Что такое сортировка сети?

пузырьковая сортировка 16 пунктов



Количество сравнения предметов в данном случае является:

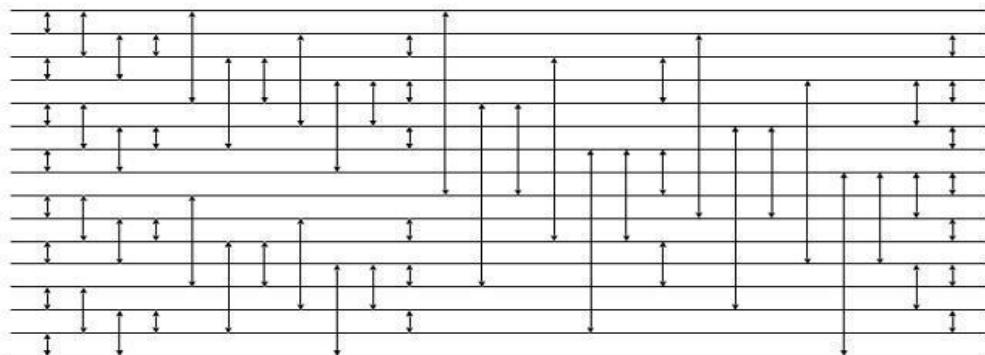
$$15 + 14 + 13 + \dots + 2 + 1 = 120$$

Тогда, что бы минимальное количество сравнения?

- ★ 65 сравнения Bose и Нельсон (1962).
- ★ 63 по Дозаторы и Флойдом и Кнута (1964).
- ★ 62 Шапиро (1969)
- ★ 60 Грин (1969)

Исх. Сортировка алгоритм Кнута и др по. аль (1964)

63 Сравнения



2. Возможный алгоритм сортировки по эволюции

Например. Целое хромосома сортировать предметы 16

140 генов, каждый из принимает целое число от 1 до 16,
позволяющих перекрытий

(12 01 05 04 16 12 04 14 01 02 06 07 15 08 10)

Какой сравнивает 12 \Leftrightarrow 01 05

\Leftrightarrow 04 16

\Leftrightarrow 12

.....

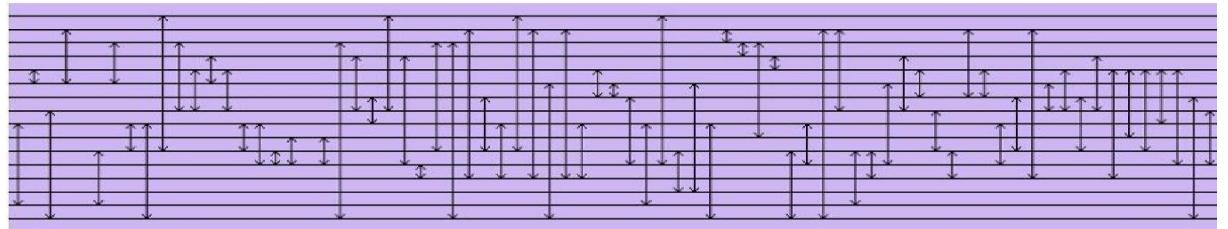
08 \Leftrightarrow 10

Простая реализация

(1) Создать одну хромосому

чтобы посмотреть, как он будет сортировать 16 целых чисел от 1 до 16 с случайнм порядкe также привлечь его графическое представление

```
chromosome: [ 08 14 04 05 15 07 05 01 07 07 10 14 02 05 08 10 08 15 10 00 02 07 07 04 03 05 04 07 08 10 08 11
11 10 11 09 03 03 09 11 15 02 07 03 06 08 07 00 11 03 12 11 02 10 02 15 12 01 06 10 08 12 00 10 01 12 15 05 01
12 12 08 06 04 06 05 06 11 14 08 00 11 13 10 05 13 08 15 01 02 03 02 02 09 03 04 15 10 08 11 01 15 01 07 10 14
10 12 05 11 03 07 04 06 10 07 10 12 01 06 06 04 11 08 06 10 12 01 07 05 04 07 10 06 03 07 12 04 04 09 10 04 04
08 11 04 15 06 07 11 ]
```



(2) Применить свой хромосому в целое число от 1 до 16 с случайному порядке

Такие как

I.

```
before [ 02 15 01 14 06 09 11 00 03 10 05 08 13 07 12 04 ]
after [ 02 00 06 01 04 03 09 05 10 08 13 11 14 07 12 15 ]
fitness = 101
real compares = 73
```

II.

```
before [ 01 04 03 05 02 13 08 11 00 10 06 07 14 09 15 12 ]
after [ 01 03 04 00 05 02 10 06 08 07 11 12 14 09 15 13 ]
fitness = 104
real compares = 73
```

III.

```
before [ 05 15 09 13 02 08 04 00 06 12 10 03 07 11 01 14 ]
after [ 03 00 06 01 04 02 09 07 05 08 12 14 13 11 10 15 ]
fitness = 100
real compares = 73
```

IV.

```
before [ 14 04 02 03 10 08 06 11 09 12 05 01 13 00 07 15 ]
after [ 05 02 03 01 04 00 10 07 06 12 11 13 14 08 09 15 ]
fitness = 96
real compares = 73
```

V.

```
before [ 12 08 13 00 03 02 07 10 01 06 14 11 15 04 05 09 ]
after [ 05 03 00 01 04 02 09 08 07 06 10 12 15 11 14 13 ]
fitness = 100
real compares = 73
```

Или, другой пример см

chromosome:															
11	02	10	12	03	01	07	03	12	06	08	03	13	06	15	
09	05	11	15	02	06	05	09	12	05	01	01	04	04	10	
15	10	05	03	07	02	11	08	09	01	03	13	15	05	04	
09	11	12	06	12	10	04	15	11	10	08	13	11	13	14	
03	04	12	08	08	14	01	15	10	01	15	09	01	09	13	
08	09	08	03	00	02	13	13	03	05	15	05	00	06	00	
06	04	12	14	14	14	12	15	14	02	14	02	04	12	06	
08	10	04	05	08	12	05	13	09	06	03	12	11	06	04	
08	01	04	09	03	15	12	11	14	05	05	00	12	15	13	
03	09	02	02	15	04	13	03	14	09	07	08	03	00	05	

(3) Фитнес-оценивается как

$fitness = 0$
FOR i=1 TO n
 FOR j=1 TO n
 IF $x(i) > x(j)$ THEN replace $x(i)$ with $x(j)$ and $fitness++$
 ELSE do nothing

Например, предположим, $N = 5$

01 03 02 05 04 \Rightarrow Фитнес = $4 + 2 + 0 + 0 = 6$ 04 02 05 01 03 \Rightarrow Фитнес = $1 + 2 + 0 + 1 = 4$ 05 01 03 02 04 \Rightarrow Фитнес = $0 + 3 + 1 + 1 = 5$ 01 02 03 05 04 \Rightarrow Фитнес = $4 + 3 + 2 + 0 = 9$

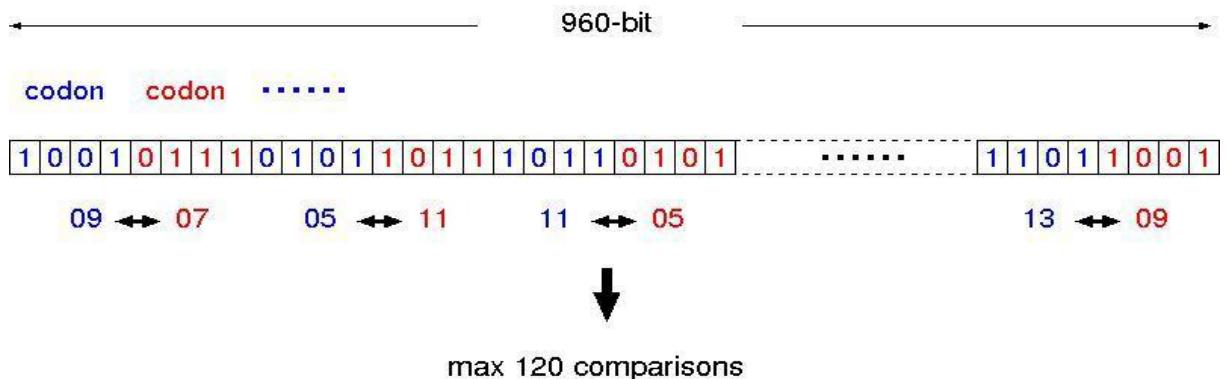
Алгоритм

- 1. Создайте N - множество целых чисел случайных порядка от 1 до 16.*
- 2. Создать население, скажем, 20 хромосом наугад.*
- 3. Оценка фитнес каждой хромосомы, применяя его к N .*
- 4. Создайте нового поколения от выбора, кроссовер и мутация.*
- 5. Повторить от 3 до 4. недо максимальных фитнес насыщения.*

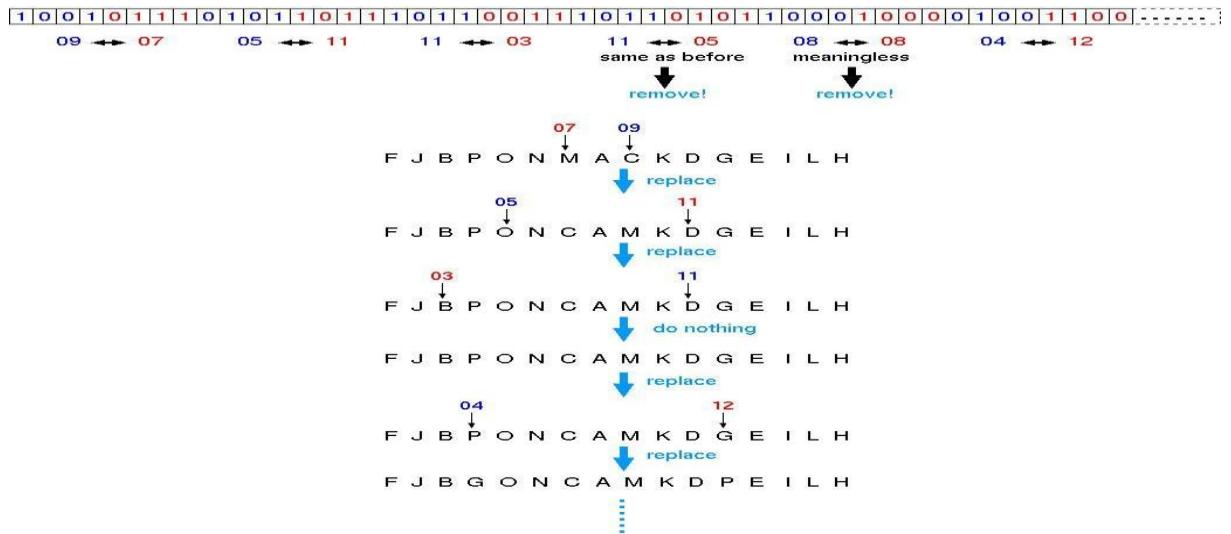
Показать (я) лучший хромосома в окончательном поколения; (II) В результате применения его в N с оригинальным того (до и после); и (III) его схема по эти идентичные сравнения.

3. Давайте использовать двоичные хромосомы

Бинарные хромосома состоит **codon** работает



Как это хромосома сортирует письма в 16 алфавитном порядке?



4. Более биологическая **диплоидная хромосома**

Хиллис диплоидная хромосома (1992)

(1001 1000 1010 1101 1110 0100 1110 0011)
 (1001 1000 1010 1101 1110 0100 1110 0011)

(1001 1000 1010 1101 1110 0100 1110 0011)
 (1001 1000 1010 1101 1110 0100 1110 0011)

(1001 1000 1010 1101 1110 0100 1110 0011)
 (1001 1000 1010 1101 1110 0100 1110 0011)

(1001 1000 1010 1101 1110 0100 1110 0011)
 (1001 1000 1010 1101 1110 0100 1110 0011)

(1001 1000 1010 1101 1110 0100 1110 0011)
 (1001 1000 1010 1101 1110 0100 1110 0011)

(1001 1000 1010 1101 1110 0100 1110 0011)
 (1001 1000 1010 1101 1110 0100 1110 0011)

(1001 1000 1010 1101 1110 0100 1110 0011)
 (1001 1000 1010 1101 1110 0100 1110 0011)

(1001 1000 1010 1101 1110 0100 1110 0011)
 (1001 1000 1010 1101 1110 0100 1110 0011)

(1001 1000 1010 1101 1110 0100 1110 0011)
 (1001 1000 1010 1101 1110 0100 1110 0011)

(1001 1000 1010 1101 1110 0100 1110 0011)
 (1001 1000 1010 1101 1110 0100 1110 0011)

(1001 1000 1010 1101 1110 0100 1110 0011)
 (1001 1000 1010 1101 1110 0100 1110 0011)

(1001 1000 1010 1101 1110 0100 1110 0011)
 (1001 1000 1010 1101 1110 0100 1110 0011)

(1001 1000 1010 1101 1110 0100 1110 0011)
 (1001 1000 1010 1101 1110 0100 1110 0011)

(1001 1000 1010 1101 1110 0100 1110 0011)
 (1001 1000 1010 1101 1110 0100 1110 0011)

(1001 1000 1010 1101 1110 0100 1110 0011)
 (1001 1000 1010 1101 1110 0100 1110 0011)

15 пар из 32-битных хромосом

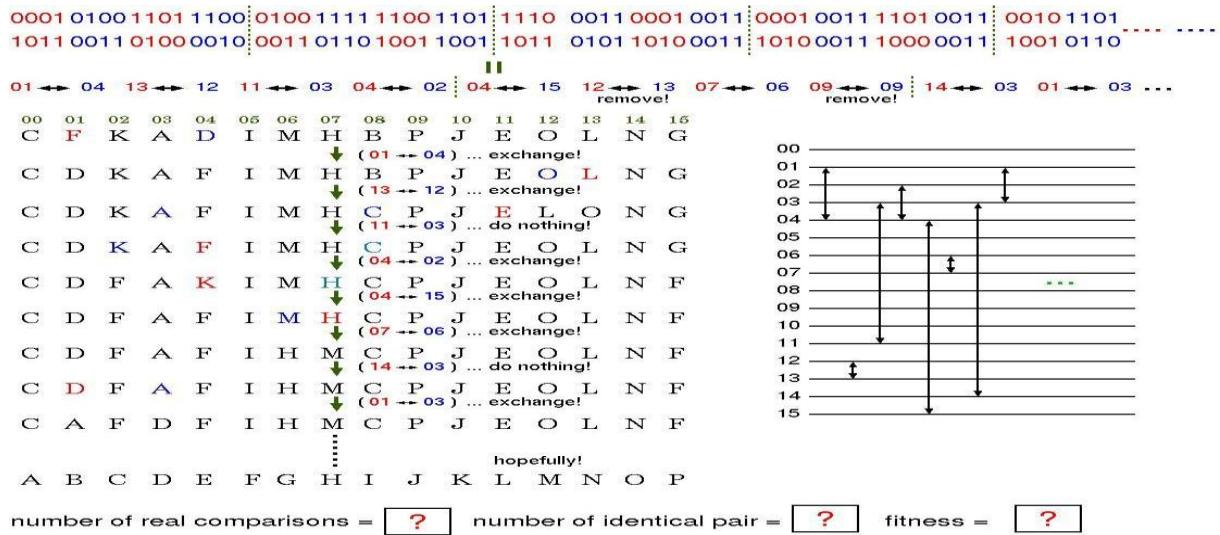
Каждая хромосома состоит из восьми 4-битовых строк, называемых кодонами каждого codon представляет собой целое число от 0 до 15

↓

60 сравнения: мировой рекорд, до сих пор!

(нет доказательств этого минимума)

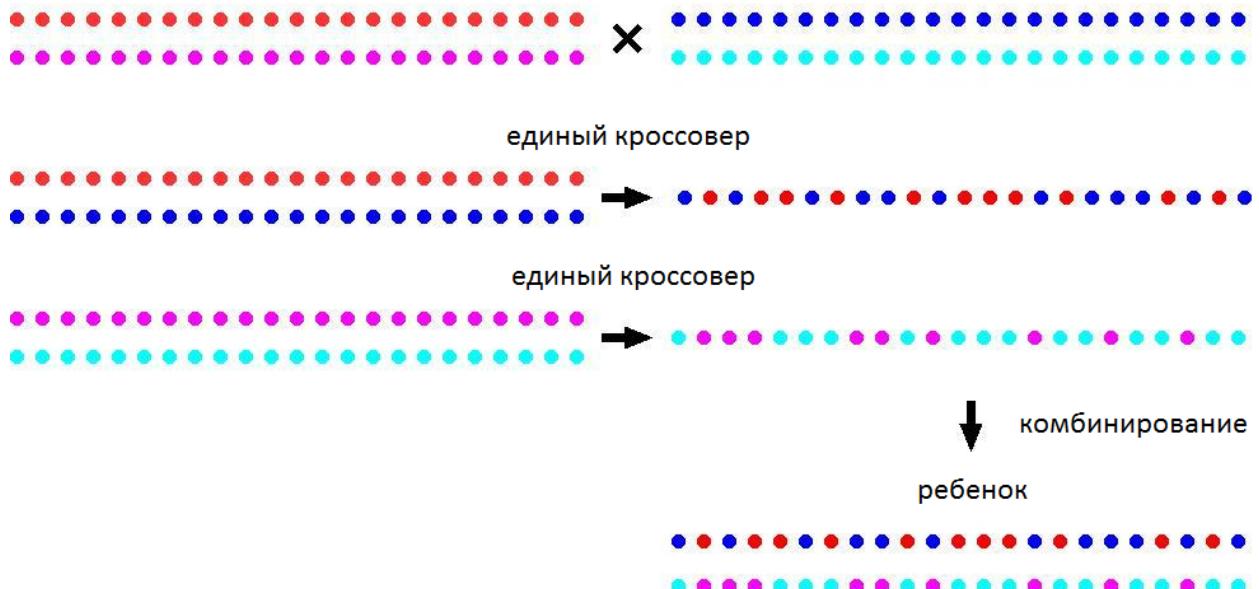
Как пара хромосом друг с 480-бит сортирует элементы 16?



Диплоидная хромосома & это кроссовер

родитель_1

родитель_2



Пребывание - Эволюция диплоидных хромосом

(1) Создать свой целевой случайную строку заказа 16 алфавитов от А до Р

(2)

Заметим, эволюцию от популяции случайных 40 диплоидных хромосом

до максимального фитнес не изменится

То, что должно быть показано:

- (1) Оригинальная строка с 16 алфавитов в случайном порядке;
- (2) Результат строки после сортировки по окончательном лучших хромосомы;
- (3) Лучшее хромосома в окончательном поколения;
- (4) Схема сравнения лучших хромосомы; **после удаления нескольких сравнений и сопоставлений само-сравнений;**
- (5) (я), что является его стоимость фитнеса? И (II), сколько в реальном сравнения включает в себя?
- (6) Три графика в процессе эволюции



Лучший хромосома может сортировать другую случайную строку?

Может лучше хромосома, фитнес которой был, как оригинал целевая строка из 16 букв, будут иметь возможность сортировать еще один новый случайную строку правильно?

Пребывание - Эволюция диплоидной хромосом сортировать несколько строк

- (1) Создание двух целевых случайном порядке 16 алфавитов от A до P.
 - Фитнес этот раз сумма, как он сортирует эти две строки.
- (2) Заметим, эволюцию от популяции случайных 40 диплоидии хромосом до максимального фитнес не изменится.
- (3) Нанесите лучший хромосому до 10 случайные строки в том числе оригиналную два!

↓

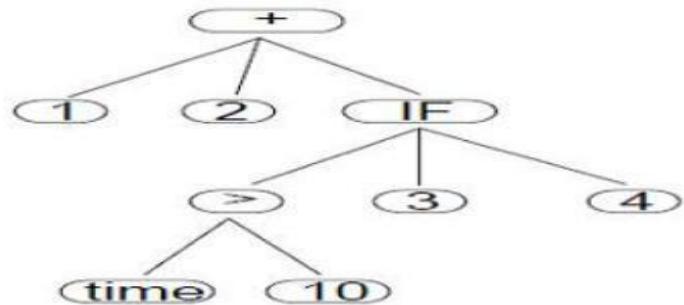
Показать набор из 10 строк с ее ДО ПОСЛЕ &!

15. Генетическое программирование

Программирование эволюции

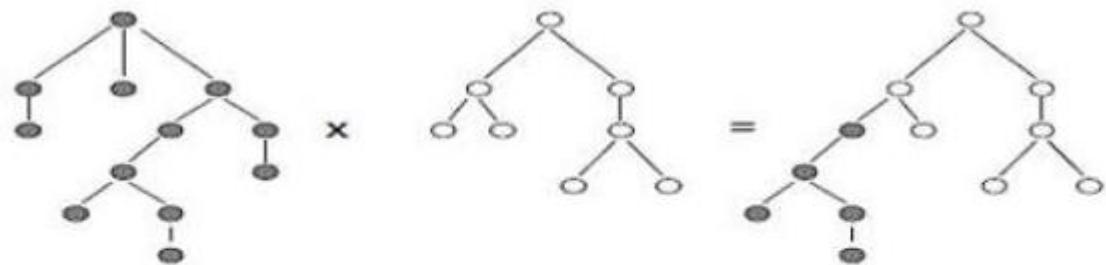
Programming в LISP, который может быть представлен деревом
такие как

```
(+ 1 2 (IF (> time 10) 3 4))
```



Это кроссовер и мутация

кроссовер



мутация

