

Генетические алгоритмы

Наумов Д.А., доц. каф. КТ

Экспертные системы и искусственный интеллект, 2019

Содержание лекции

1 Эволюционные вычисления

Эволюционные вычисления

общий термин для описания алгоритмов поиска, оптимизации, машинного обучения или анализа данных, основанных на некоторых формализованных принципах естественного эволюционного отбора и генетики.

Основные парадигмы:

- генетические алгоритмы – двоичное и вещественное представление решений-хромосом
- генетическое программирование – программы, деревья графы как представления решений-хромосом
- эволюционные стратегии – вещественные вектора или скаляры как представления решений-хромосом
- эволюционное программирование – конечный автомат как представления решений-хромосом

Три основных принципа эволюционного подхода

Charles Darwin, 1859.

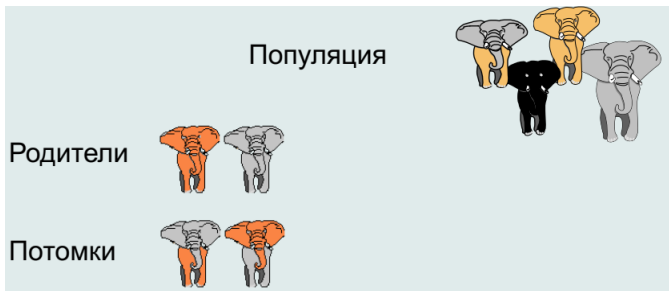
Принцип выживания сильнейших и естественный отбор



Три основных принципа эволюционного подхода

Мендель, 1865.

Основной принцип механизма наследования – хромосомы потомков состоят из частей хромосом их родителей



Три основных принципа эволюционного подхода

де Вре, 1900

Принцип мутации – существенные (разные) изменения свойств потомков и приобретение ими свойств, отсутствующих у родителей

До мутации



После мутации



Генетические алгоритмы

случайно направленные поисковые алгоритмы, которые эмулируют процесс естественной эволюции для построения (суб)оптимального решения задачи.

- реализуют принцип «выживание сильнейших» среди рассматриваемых структур;
- формирует и изменяет поисковый алгоритм на основе моделирования эволюции

Некоторые основатели и исследователи: J.Holland, D.Goldberg, Z.Michalewicz

Генетические алгоритмы

Генетический алгоритм характеризуется следующими основными параметрами:

$$= P^0, \lambda, l, s, p, f, t$$

где

- $P^0 = (X_1^0 \dots X_\lambda^0)$ - исходная популяция;
- X_1^0 - решение задачи, представленное в виде хромосомы;
- λ - размер популяции
- l - длина каждой хромосомы;
- s - оператор отбора;
- p - отображение, определяющее рекомбинацию;
- f - целевая функция;
- t - критерий остановки алгоритма.

Блок-схема классического генетического алгоритма



Кодирование хромосомы

Хромосома

потенциальное решение представляется двоичной строкой – искусственной хромосомой, где каждый «ген» равен 0 или 1.



Популяция

множество потенциальных решений. Развивается по законам искусственной эволюции на основе генетических операторов

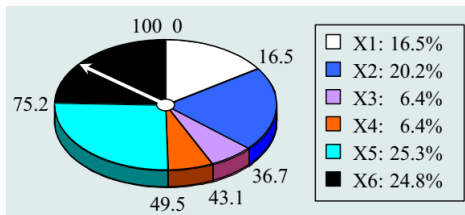
Кодирование хромосомы

Двоичные строки	(0101 ... 1100)
Действительные числа	(43.2 -33.1 ... 0.0 89.2)
Перестановки элементов	(E11 E3 E7 ... E1 E15)
Списки правил	(R1 R2 R3 ... R22 R23)
Элементы программ	(genetic programming)
Матрицы	
Графы	
.. любые структуры данных...	

Оператор селекции

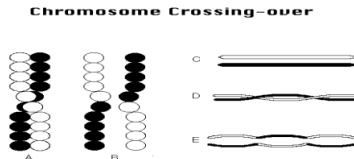
Ассиметричное колесо рулетки

Популяция отображается на колесо рулетки, где каждая особь представляется сектором, размер которого пропорционален значению фитнес функции.



Особи выбираются путем повторного вращения на основе стохастического выбора с возвращением.

Оператор кроссинговера



Одноточечной или простой оператор кроссинговера выполняется с вероятностью P_c в 2 этапа:

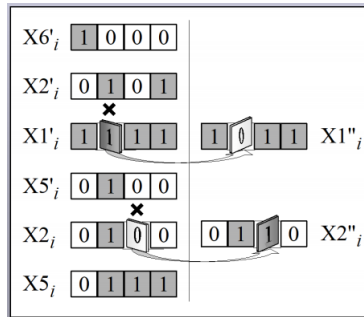
- 1) целое число – точка кроссинговера k ($1 \leq k \leq n-1$) выбирается случайным образом для выбранных строк $A=a_1 a_2 \dots a_n$ и $B=b_1 b_2 \dots b_n$;
- 2) особи A и B обмениваются подстроками после позиции k и производят две новые особи $A=a_1 a_2 \dots a_k b_{k+1} \dots b_n$ и $B=b_1 b_2 \dots b_k a_{k+1} \dots a_n$.

010		010100		010хухуух		010		01		0100		010ху0100
хху		хухуух	\Rightarrow	хху010100		хху		ху		хуух	\Rightarrow	хху01хуух

Оператор мутации

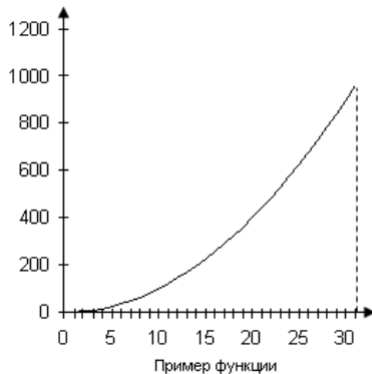
Оператор мутации случайным образом (с малой вероятностью P_m) изменяет значения некоторых генов особей популяции:

01001011 -> 01000011



Пример

Найти (для простоты) целочисленное значение x на отрезке от $[0, 31]$, при котором функция $f(x) = x^2$ принимает максимальное значение.



Репродукция

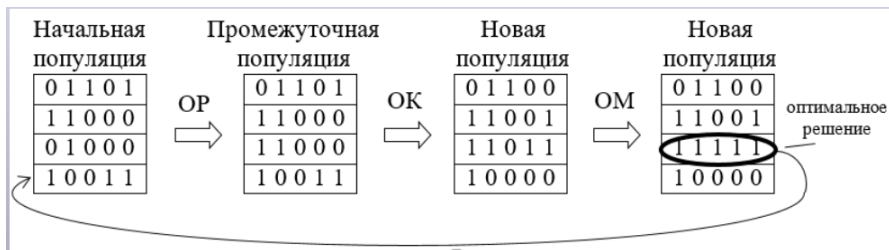
№ хромосомы	Начальная популяция особей	Десятичное значение x	Значение $f(x)$	$\frac{f(x_i)}{\sum f(x_j)}$	Среднее значение $\bar{f}(x)$	Максимальное значение $f_{max}(x)$
1	01101	13	169	0,14	293	576
2	11000	24	576	0,49		
3	01000	8	64	0,06		
4	10011	19	361	0,31		

Кроссинговер

№ хромосомы	Популяция после репродукции	Пары хромосом для кроссинговера	Популяция после кроссинговера	Значение $f(x)$	Среднее значение $\bar{f}(x)$	Максимальное значение $f_{max}(x)$
1	0 1 1 0 1 ↑	1-2	01100	144	439	729
2	1 1 0 0 0 ↑	1-2	11001	625		
3	1 1 0 0 0 ↑	3-4	11011	729		
4	1 0 0 1 1 ↑	3-4	10000	256		

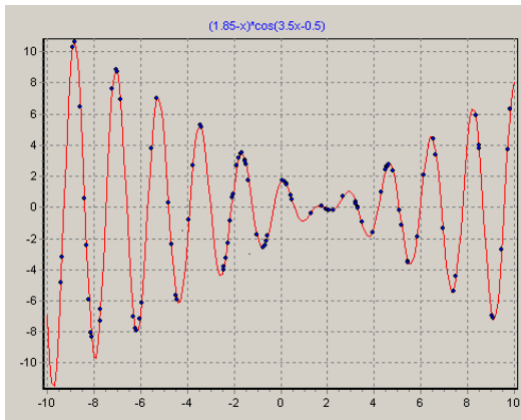
Мутация

№ хромосомы	Популяция после кроссинговера	Новая популяция после мутации	Десятичное значение x	Значение $f(x)$	Среднее значение $\bar{f}(x)$	Максимальное значение $f_{max}(x)$
1	01100	01100	12	144	496.5	961
2	11001	11001	25	625		
3	11011	11111	31	961		
4	10000	10000	16	256		



Пример

Рассмотрим $f(x) = (1,85 - x) * \cos(3,5x^{0,5})$, необходимо найти вещественное x , которое максимизирует f



Пример функции с популяцией особей в начале эволюции

Кодирование решения

- Для представления вещественного решения (хромосомы) будем также использовать двоичный вектор, который применяется в классическом простом ГА.
- Его длина зависит от требуемой точности решения, которую в данном случае положим 3 знака после запятой.
- Поскольку отрезок области решения имеет длину 20, для достижения заданной точности отрезок $[a,c] = [-10,+10]$ должен быть разбит на равные части (маленькие отрезки), число которых должно быть не менее $20 \cdot 1000$.
- В качестве двоичного представления используем двоичный код номера (маленького) отрезка. Этот код позволяет определить соответствующее ему вещественное число, если известны границы области решения

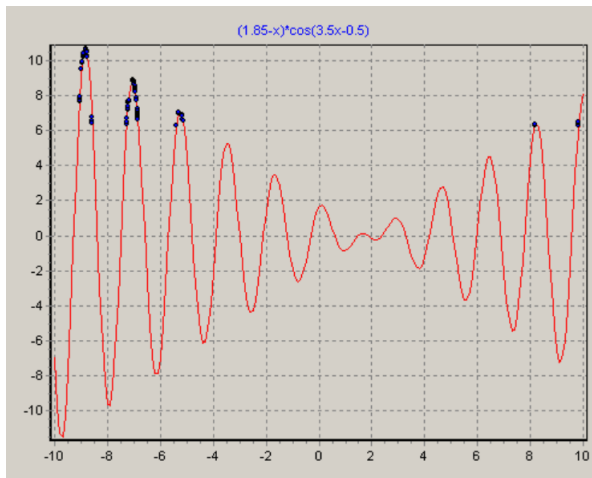
Кодирование решения

Отображение из двоичного представления в вещественное число из отрезка:

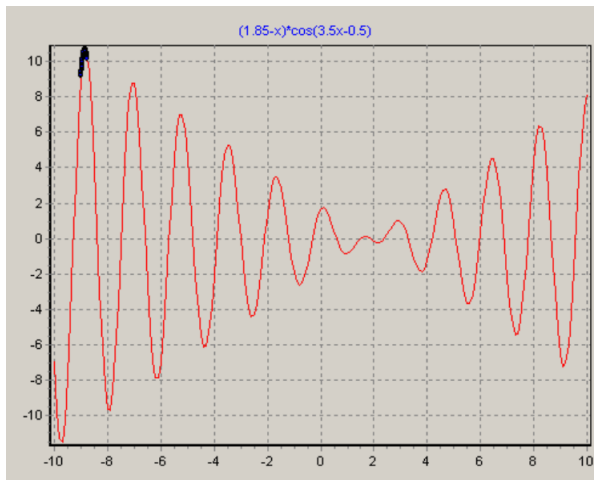
$$x = a + x' \cdot \frac{(c - a)}{2^{15} - 1} = -10 + x' \cdot \frac{20}{2^{15} - 1}$$

Хромосомы (0000000000000000) и (1111111111111111) представляют границы отрезка -10 и $+10$ соответственно.

Работа ГА

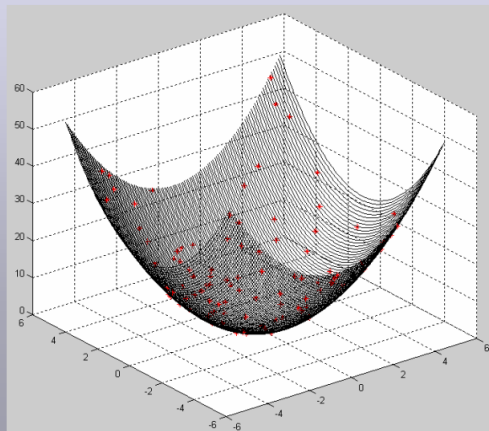


Работа ГА

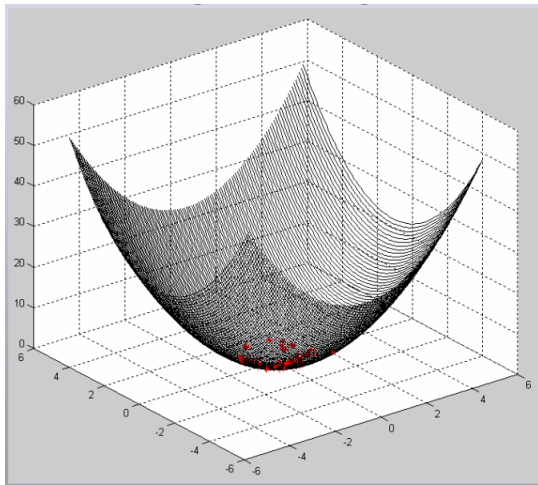


Пример оптимизации функции двух переменных

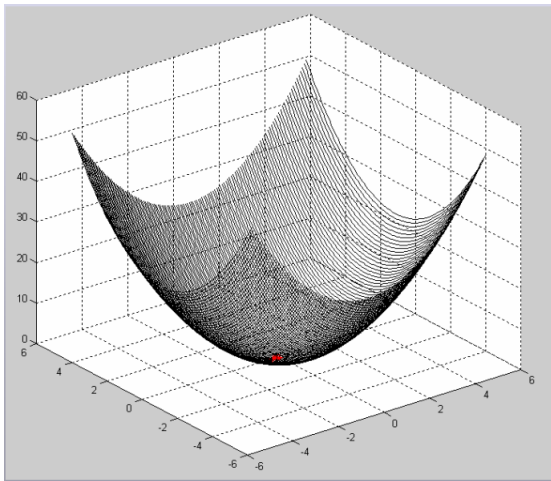
$$\min[F(x_1, x_2) = x_1^2 + x_2^2]$$



Пример оптимизации функции двух переменных



Пример оптимизации функции двух переменных

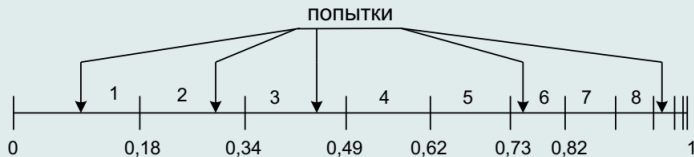


Оператор селекции: рулетка

$$P^t \xrightarrow{P_s(a_i^t)} \tilde{P}$$

1) roulette wheel selection $P_s(a_i^t) = \frac{f(a_i^t)}{\sum_{j=1}^N f(a_j^t)}$

Ind. number	1	2	3	4	5	6	7	8	9	10	11
FF value	2,0	1,8	1,6	1,4	1,2	1,0	0,8	0,7	0,3	0,2	0,0
Selection probability	0,18	0,16	0,15	0,13	0,11	0,09	0,07	0,06	0,03	0,02	0,00



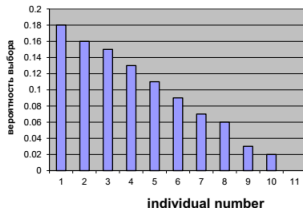
Оператор селекции: линейный ранк

2) linear rank-based selection
$$P_s(a_i^t) = \frac{1}{N} \left(\alpha - (\alpha - \beta) \frac{i-1}{N-1} \right)$$

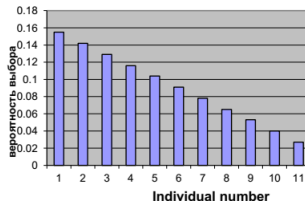
where parameter $1 \leq \alpha \leq 2$ is chosen randomly, $\beta = 2 - \alpha$

Ind. number	1	2	3	4	5	6	7	8	9	10	11
FF value	2	1,8	1,6	1,4	1,2	1	0,8	0,7	0,3	0,2	0
Selection probability	0,155	0,142	0,129	0,116	0,104	0,091	0,078	0,065	0,053	0,04	0,027

Individuals ranking in selection by asymmetric roulette wheel model

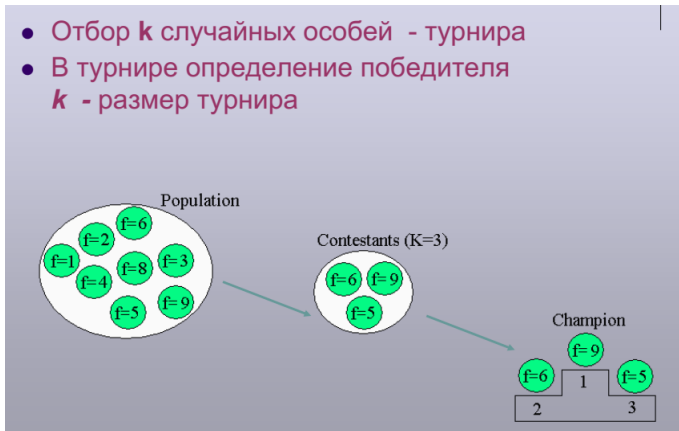


Individuals ranking in selection by linear ranking method with equal step
 $a=1,7$ $b=0,3$



Оператор селекции: турнирный отбор

- Отбор k случайных особей - турнира
- В турнире определение победителя
 k - размер турнира



Пример: дискретная рекомбинация



Пример: дискретная рекомбинация


Двоичный алфавит:

Genotype:

1	0	1	0	0	0	1	1
---	---	---	---	---	---	---	---

Phenotype:

= 163



$$1 \cdot 2^7 + 0 \cdot 2^6 + 1 \cdot 2^5 + 0 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0$$

=

$$128 + 32 + 2 + 1 = 163$$


Пример: дискретная рекомбинация

Действительное число:

Genotype:

1	0	1	0	0	0	1	1
---	---	---	---	---	---	---	---

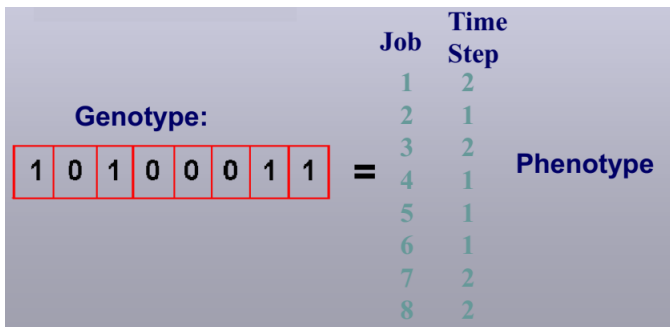
Phenotype: **= 13.9609**



$$x = 2.5 + \frac{163}{256} (20.5 - 2.5) = 13.9609$$

Пример: дискретная рекомбинация

Фенотип как очередь:



Пример: дискретная рекомбинация

Действительное представление в ввиде массива чисел:

- Особи представляются векторами действительных чисел

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, x_i \in R$$


- Фитнесс- функция :

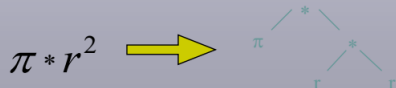
Отображает

$$f: R^n \rightarrow R$$

Пример: дискретная рекомбинация

Древовидное представление:

- Особи популяции - деревья. 
- Любое S-выражение может быть представлено деревом функций и термина
 - Функции: sine, cosine, add, sub, and, If-Then-Else, Turn...
 - Терминалы: X, Y, 0.456, true, false, π , Sensor0...
- Пример: вычисление площади окружности:



Оператор мутации

- Операторы мутации должны позволять исследовать пространство решений
- Важен размер шага мутации, желательно его контролировать
- Мутация должна производить правильные решения

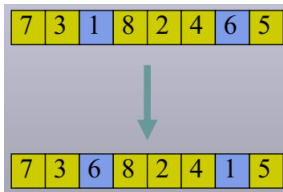


Мутация для действительного представления

- Выполняется путем наложения случайного шума
- Часто используется Гауссово, нормальное распределение

$$x'_i = x_i + N(0, \sigma)$$

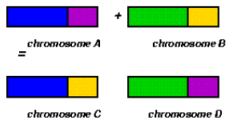
Мутация для упорядоченного представления (случайный выбор генов и обмен)



Операторы рекомбинации

- Потомок должен взять что то от каждого родителя.
- Оператор рекомбинации должен разрабатываться вместе с кодированием(представлением) решения.
- Рекомбинация должна производить правильные решения.
- Рекомбинация выполняется с вероятностью P_c .

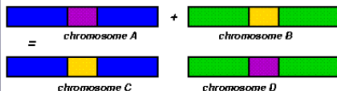
single point crossover:



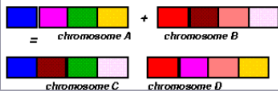
e.g.

$$11010000 + 01011100 = 11011100 \\ 01010000$$

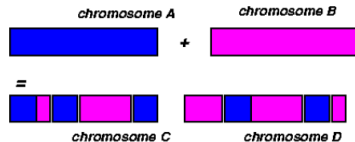
two point crossover:



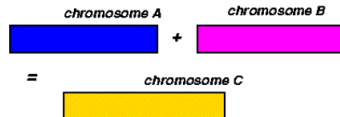
multiple point crossover:



Uniform crossover (bits randomly copied):



Arithmetic crossover:



e.g. 100101010100 AND 110111000111
=
100101000100

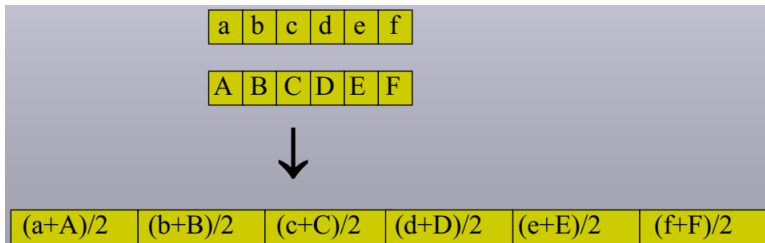
Однородный кроссинговер

Производит потомка от 2-х заданных родителей следующим образом



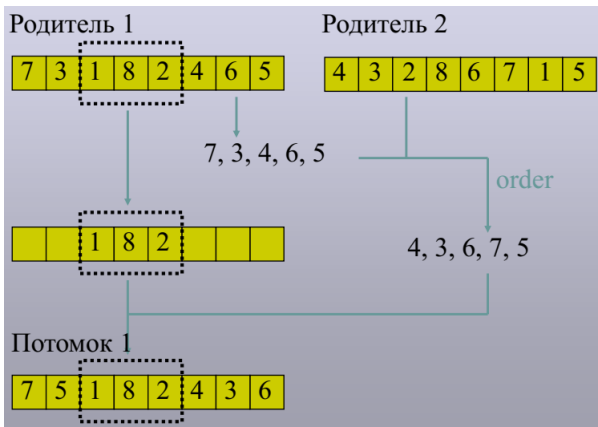
Арифметический кроссинговер

По 2-м данным родителям потомок производится следующим образом



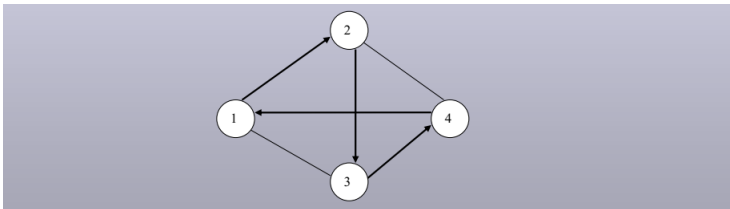
Рекомбинация для упорядоченного представления

По 2-м данным родителям потомок производится следующим образом



Задача коммивояжера

Коммивояжер (бродячий торговец) должен выйти из первого города, посетить по одному разу в некотором порядке все города и вернуться в исходный город. Расстояния между городами известны. В каком порядке следует обходить города, чтобы замкнутый путь (тур) коммивояжера был кратчайшим?



Задача коммивояжера: формальная постановка

- имеется полный взвешенный ориентированный граф без петель G с множеством вершин $N=1,2,\dots,n$;
- веса всех дуг неотрицательны;
- в этом графе требуется найти гамильтонов цикл с минимальной стоимостью.
- Исходная информация по ЗК представляется в виде матрицы - вес дуги (i,j) графа G все элементы главной диагонали нулевые (но в некоторых постановках полагаются).
- Тур коммивояжера может быть описан циклической перестановкой $t=(j_1,j_2,\dots,j_n,j_1)$, причём все $j_1..j_n$ – разные; повторяющийся в начале и в конце номер города j_1 , показывает, что перестановка циклическая.
- Пространством поиска решений этой задачи является множество перестановок n городов.
- Размерность пространства поиска задачи пропорциональна $(n-1)!$

Задача коммивояжера



Представление решения

Решение представляется списком городов

1) Berlin	3) Stuttgart	5) Cologne	7) Dusseldorf
2) Munich	4) Wesbaden	6) Hanover	8) Breme

CityList1 (3 5 7 2 1 6 4 8)

CityList2 (2 5 7 6 8 1 3 4)

Мутация



Кроссинговер

- “Обычный” кроссинговер часто порождает неправильные решения



- Поэтому разработаны «проблемно-ориентированные» операторы кроссинговера

Упорядоченный кроссинговер

- Сохраняет относительный порядок городов в туре
- Оператор:
 1. Выбор произвольного фрагмента из первого родителя
 2. Копирование этого фрагмента в первый потомок
 3. Копирование городов, отсутствующих в первом фрагменте, из второго родителя в первый потомок:
 - Начиная с правой точки кроссинговера,
 - Сохраняя порядок городов во втором родителе
 - По «кольцу»
 4. Аналогично для второго потомка, где родители меняются местами

Пример упорядоченного кроссинговера

- Копирование случайного фрагмента

1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---



			4	5	6	7		
--	--	--	---	---	---	---	--	--

9	3	7	8	2	6	5	1	4
---	---	---	---	---	---	---	---	---

- Копирование остатка из 2-го родителя в порядке 1,9,3,8,2

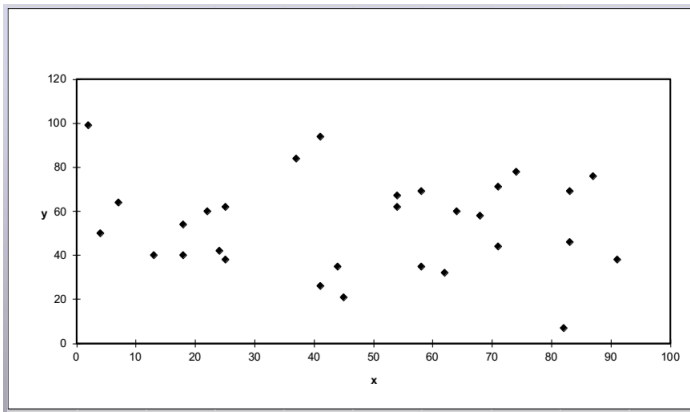
1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---



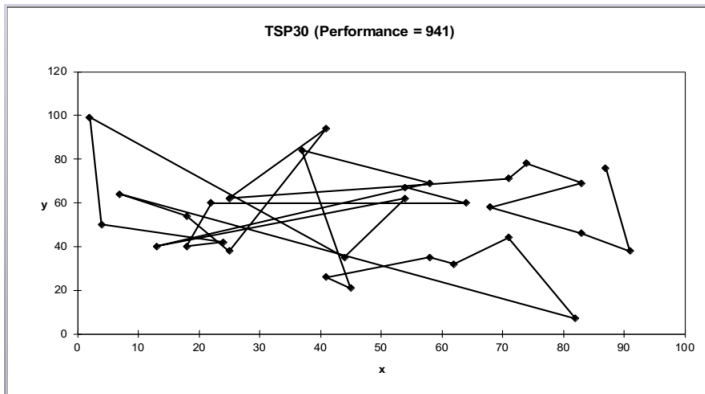
3	8	2	4	5	6	7	1	9
---	---	---	---	---	---	---	---	---

9	3	7	8	2	6	5	1	4
---	---	---	---	---	---	---	---	---

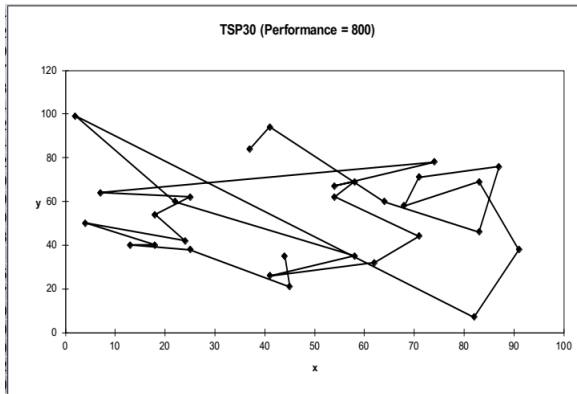
ЗК: 30 городов



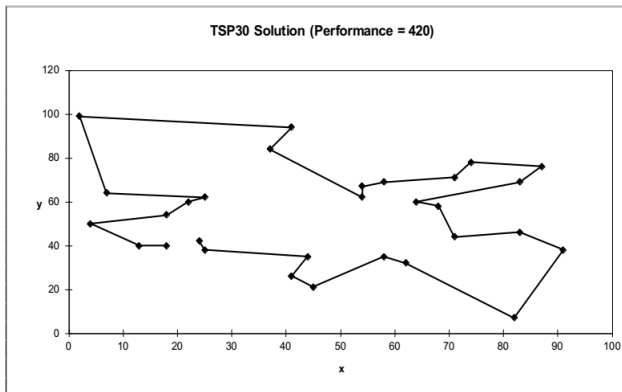
ЗК: 30 городов



ЗК: 30 городов



ЗК: 30 городов



3К: 30 городов

