

# Математическое и программное обеспечение для построения и визуализации рациональных маршрутов с использованием генетических алгоритмов

Выполнил: ст. группы МО-405

Абзалтдинов Л. И.

Руководитель: к.т.н., доц. кафедры ВМиК

Тарасова Т. Д.

# Цель и задачи

## Цель:

- Исследование эффективности генетических алгоритмов для поиска рациональных маршрутов.
- Снижение стоимости маршрута проезда по заданным объектам.

## Задачи:

- Провести обзор существующих программных решений задачи построения рационального маршрута.
- Провести обзор и анализ методов решения данной задачи.
- Разработать алгоритмы поиска рационального маршрута;.
- Разработать программное обеспечение для построения и визуализации рационального маршрута.
- Провести вычислительный эксперимент, проанализировать полученные результаты.
- Построить маршруты по реальным данным.

# Обзор программных решений

Программное обеспечение	Сложность маршрутов	Визуализация маршрута	Дополнительная функциональность	Цена
<b>Mapbox</b>	До 12 местоположений	Нет	Учитывает тип маршрута: пеший / велосипедный / автомобильный	бесплатно
<b>MAPQUEST</b>	До 25 местоположений	Нет	Выбор типа оптимизации: самый быстрый или короткий маршрут	бесплатно
<b>GraphHopper</b>	До 150 местоположений	Нет	Возможность учитывать количество используемых транспортных средств	До 239€ / месяц
<b>ПК «Логистика развоза»</b>	Позволяет составлять маршруты любой сложности	Есть	Учет грузоподъемности автомобиля	>= 50 000 руб.
<b>Муравьиная логистика</b>	Позволяет составлять маршруты любой сложности	Есть	Учет грузоподъемности автомобиля	35\$ – 500\$ (зависит от количества маршрутов в день)
<b>ANTOR LogisticsMaster™</b>	Позволяет составлять маршруты любой сложности	Есть	Учет грузоподъемности автомобиля, статистики пробок, планирование доставки с дозагрузкой	>= 200 000 руб.

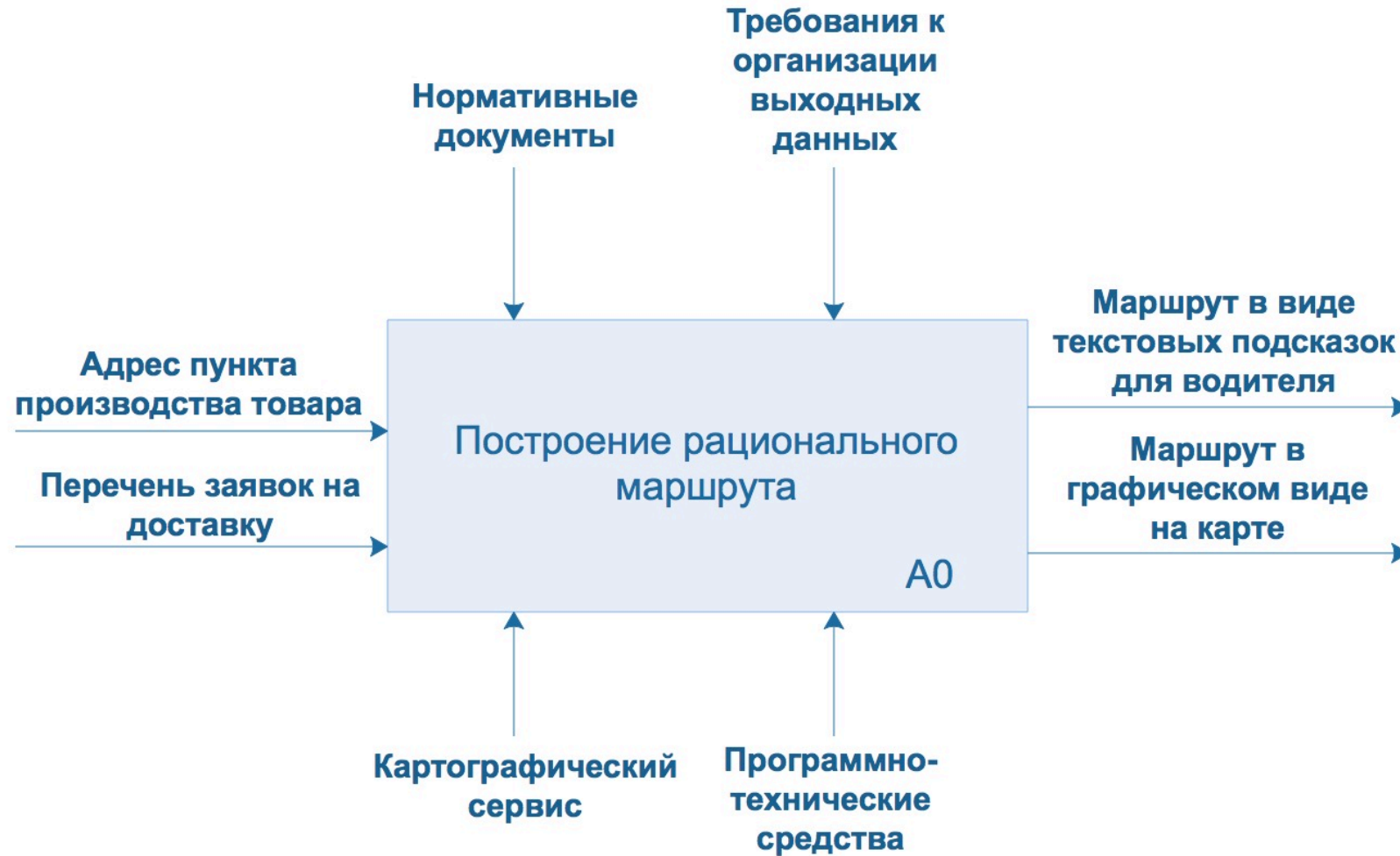
# Содержательная постановка задачи

## Дано:

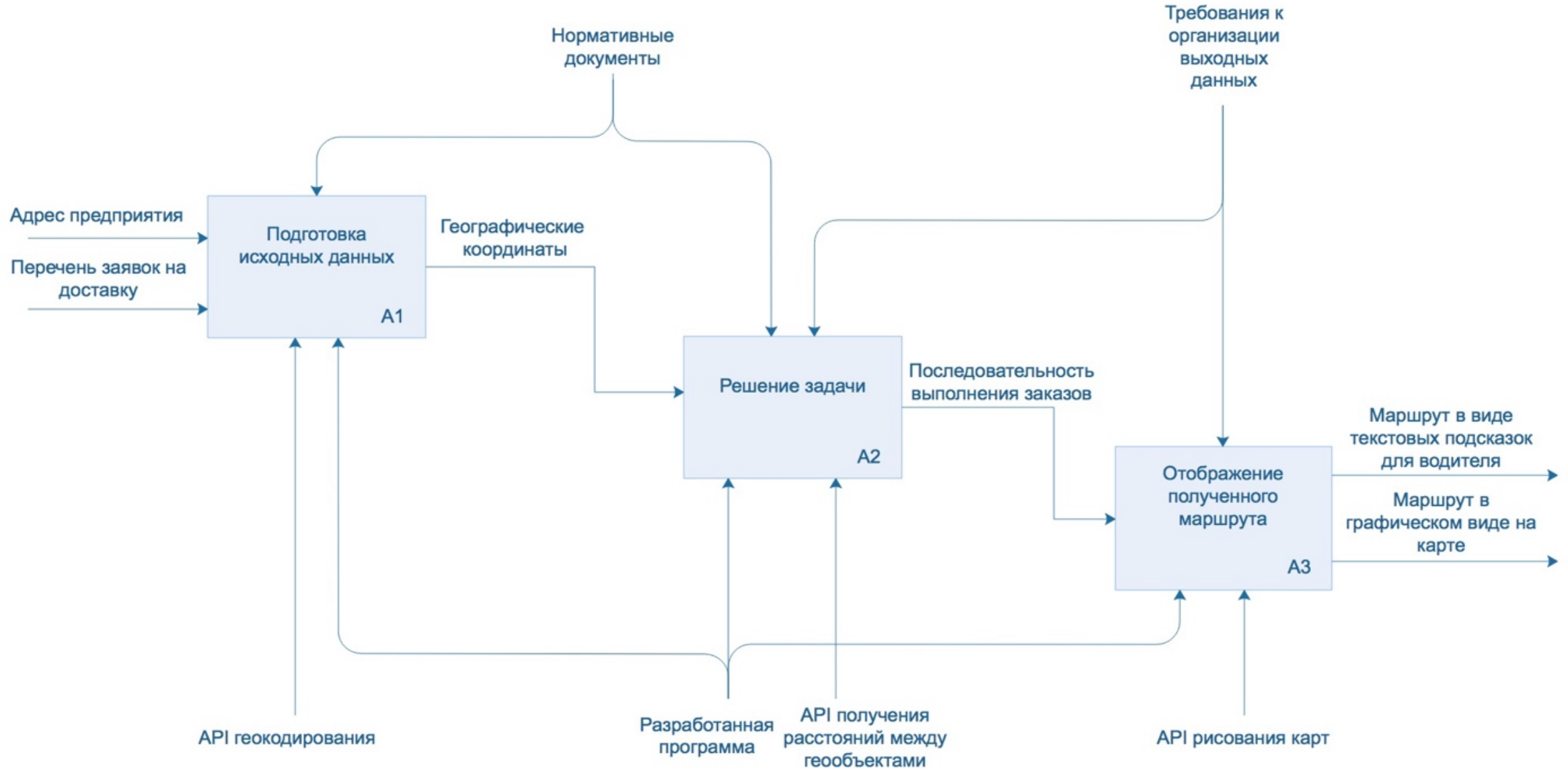
- Географический адрес расположения пункта производства товара данного предприятия
- Перечень заявок на доставку товара с указанием адреса и даты доставки.

Требуется найти маршрут минимальной стоимости на определенную дату, начинающийся и заканчивающийся в пункте производства и проходящий по всем адресам заявок на доставку ровно один раз.

# Формальная постановка задачи



# Декомпозиция задачи



# Математическая модель задачи коммивояжера

- Дано:

$N$  – количество городов,

$C = (c_{ij}), i, j = \overline{1, N}$  – матрица расстояний между городами.

$$c_{ij} \geq 0, \forall i, j; \quad c_{ii} = \infty, \forall i$$

- Найти:

Замкнутый маршрут (цикл) в виде матрицы  $X = (x_{ij}), i = \overline{1, N}, j = \overline{1, N}$ , удовлетворяющий следующим ограничениям:

1°. Маршрут должен проходить через каждый город ровно один раз:

$$\sum_{i=1}^N x_{ij} = 1, j = \overline{1, N}; \quad \sum_{j=1}^N x_{ij} = 1, i = \overline{1, N},$$

$$x_{ij} = \begin{cases} 1, & \text{если из города } i \text{ едем в город } j \\ 0, & \text{в противном случае} \end{cases}, i = \overline{1, N}, j = \overline{1, N}$$

2°. Маршрут должен иметь минимальное суммарное пройденное расстояние:

$$L = \sum_{i=1}^N \sum_{j=1}^N c_{ij} x_{ij} \rightarrow \min$$

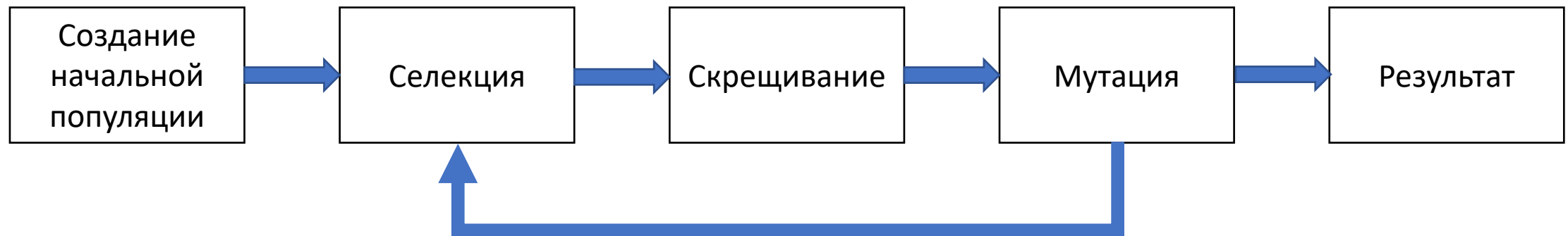
# Методы решения задачи





# Генетический алгоритм

Цикл функционирования основного этапа генетического алгоритма в рамках одного поколения особей в популяции:



Целевая функция – суммарная стоимость маршрута.

Ген – номер города

Особь – маршрут обхода

Хромосома – порядок обхода городов

Популяция – конечное множество возможных маршрутов обхода

# Способ кодирования хромосом

Для разработки генетического алгоритма сначала необходимо определить способ кодирования хромосом:

1. Двоичное кодирование:

0 – 3 – 1 – 2 – 4  
↓  
(000|011|001|010|100)

2. Представление в виде пути обхода:

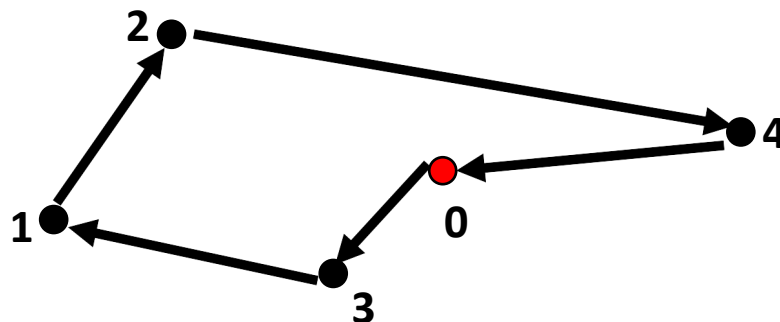
0 – 3 – 1 – 2 – 4  
↓  
(0, 3, 1, 2, 4)



3. Представление в виде списка смежности:

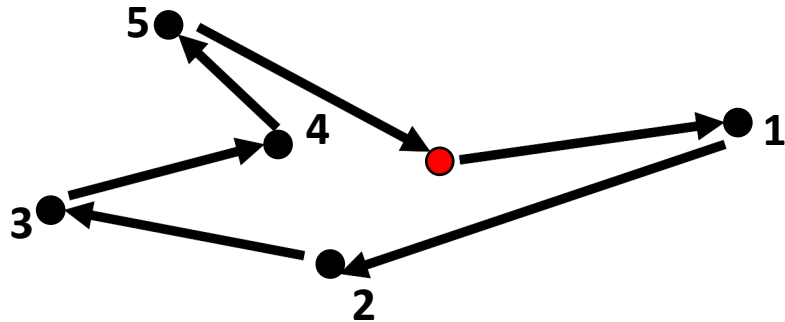
0    1    2    3    4  
0 – 3 – 1 – 2 – 4  
↓  
(3, 2, 4, 1, 0)

Пример маршрута:

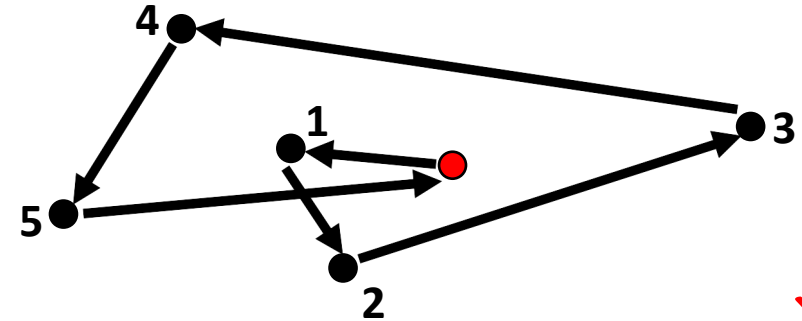


# Создание начальной популяции

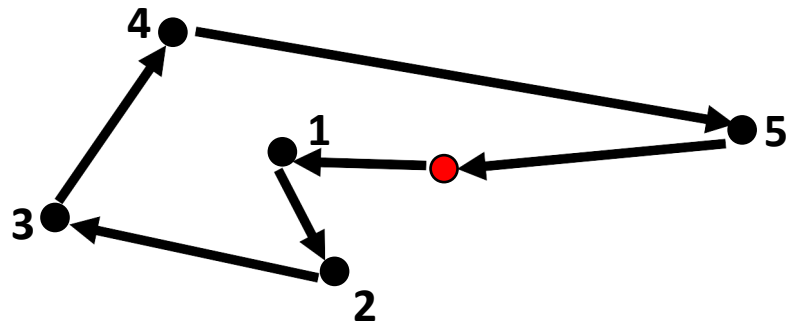
1. Случайный обход



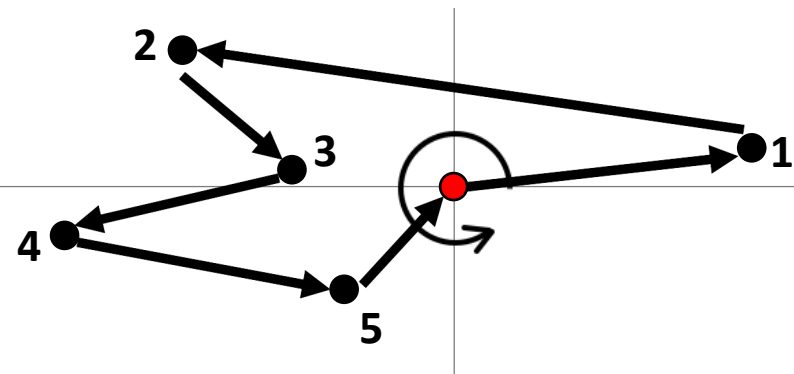
3. Обход по мере возрастания расстояния от начального пункта



2. Алгоритм ближайшего соседа



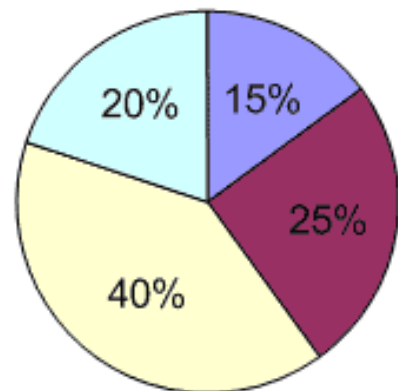
4. Обход по кругу вокруг начального пункта



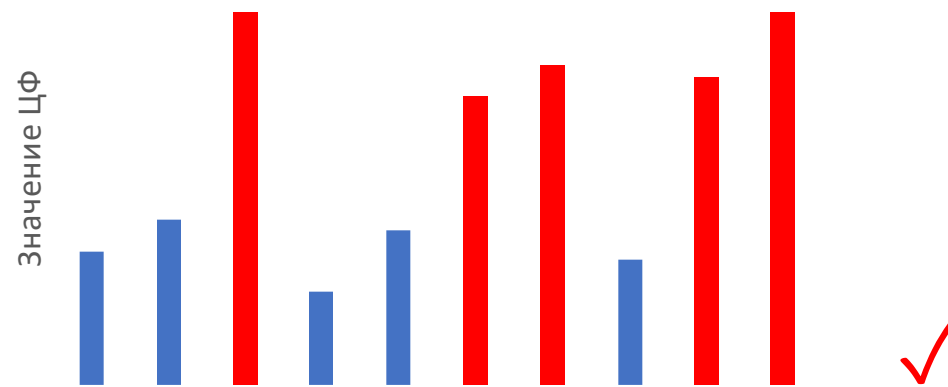
# Операторы селекции

## 1. Пропорциональная селекция

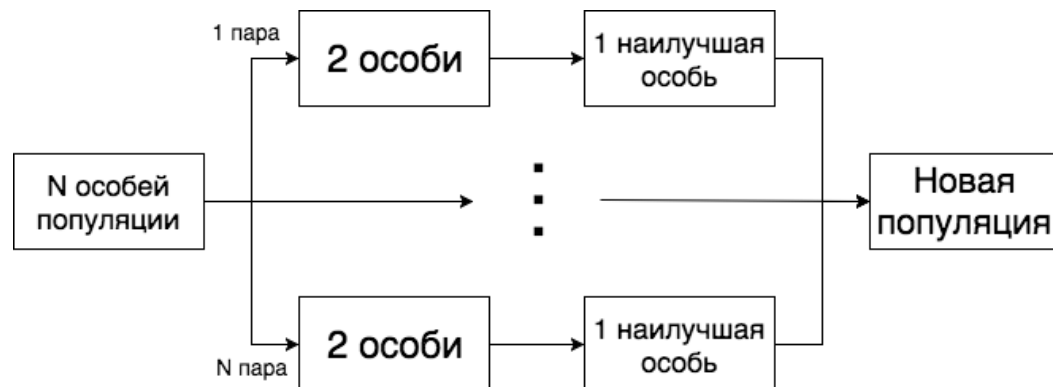
Хромосома	ЦФ	%
1 (1 - 2 - 3 - 4)	150	15%
2 (3 - 2 - 4 - 1)	250	25%
3 (2 - 3 - 1 - 4)	400	40%
4 (4 - 2 - 3 - 1)	200	20%



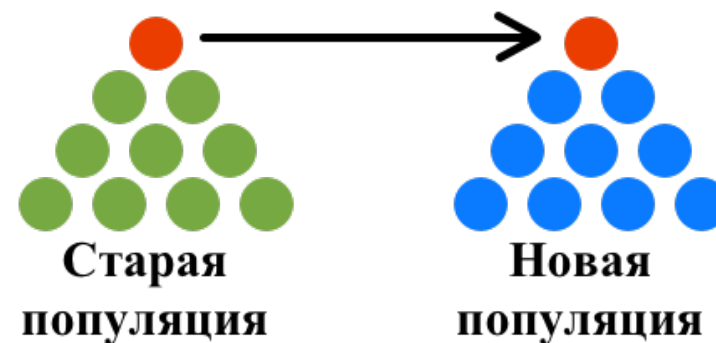
## 3. Ранговая селекция



## 2. Турнирная селекция

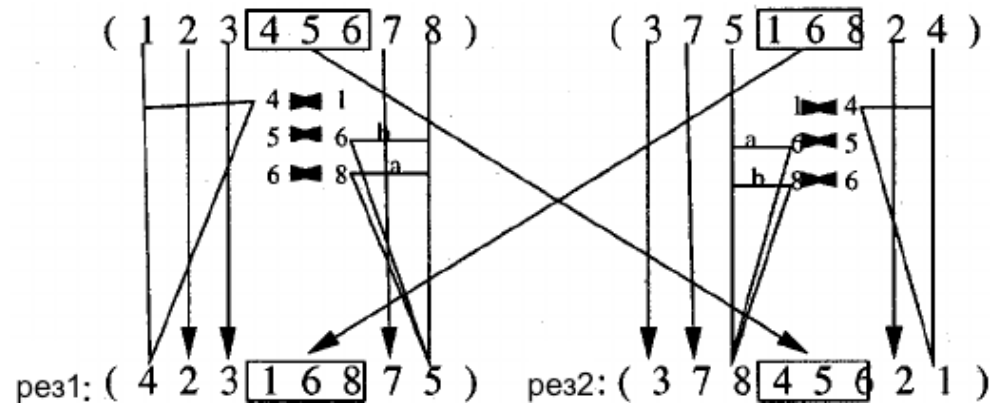


## \* Элитарная стратегия

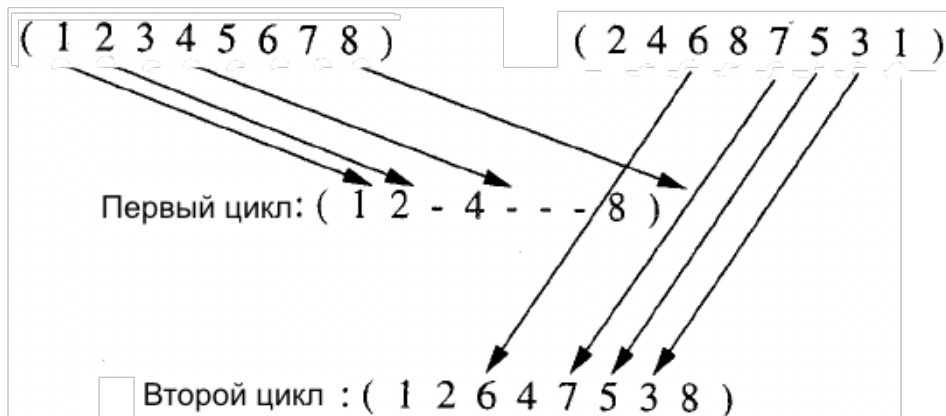


# Операторы скрещивания

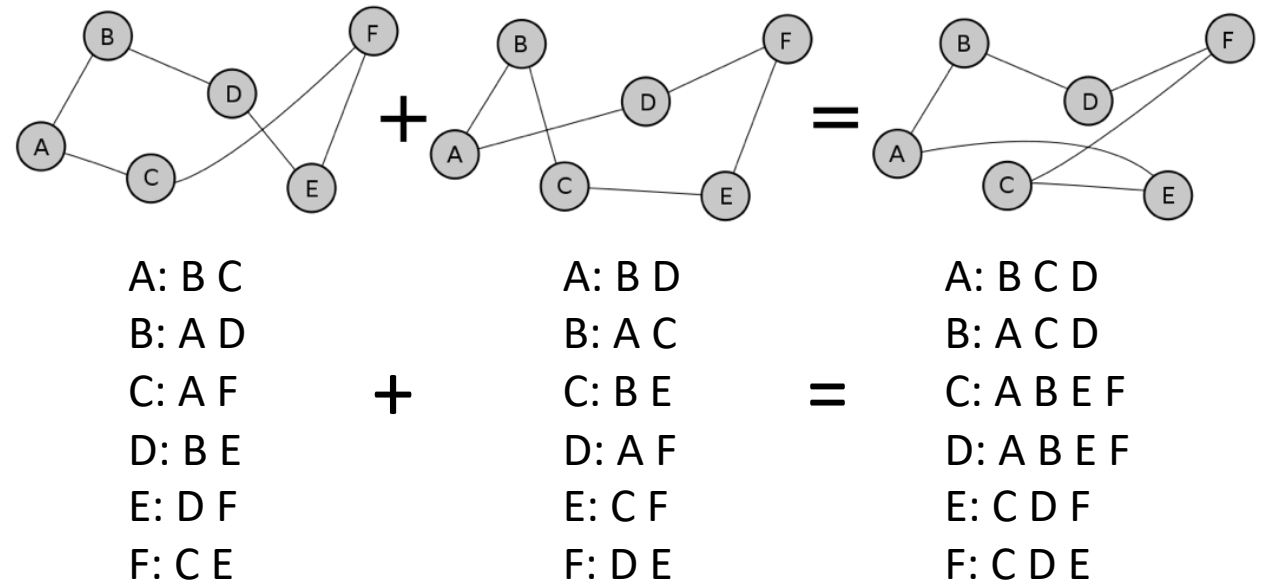
## 1. Частично отображающий кроссовер



## 2. Циклический кроссовер

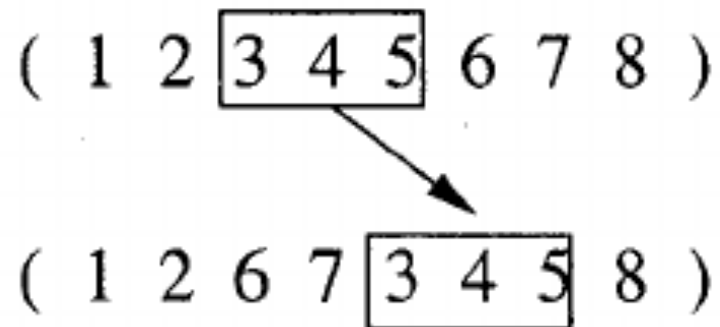


## 3. Рекомбинация дуг

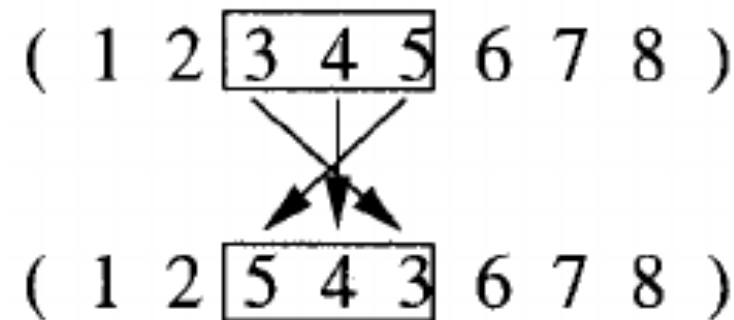


# Операторы мутации

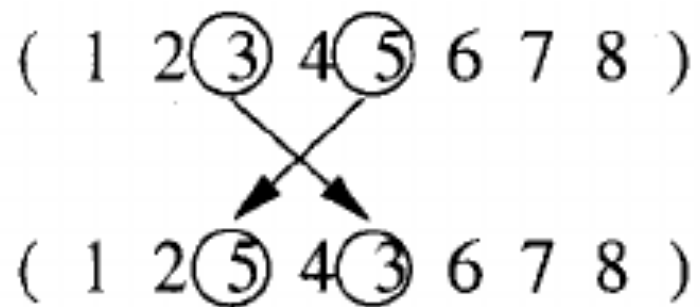
1. Перемещение последовательности генов (move)



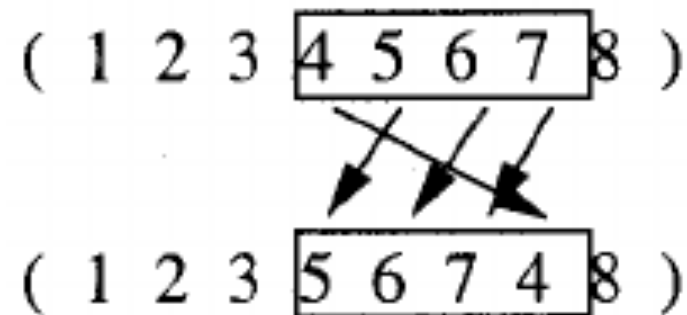
3. Разворот последовательности генов (reverse)



2. Перестановка двух генов (exchange)



4. Случайная перестановка части генов (scramble)



# Модификации генетического алгоритма

С учетом выбранных ранее компонентов генетического алгоритма создано 18 модификаций ГА комбинированием следующих параметров:

**Оператор кроссовера:**

1. С 2-мя родителями
2. С 3-мя родителями

**Оператор мутации:**

1. Перемещение (move)
2. Разворот (reverse)
3. Перемещение + разворот  
(move+reverse)

**Вероятность мутации:**

1. 0,1
2. 0,3
3. 0,5

Для удобства будем обозначать алгоритмы по следующему правилу:

$\{2;3\} \{R;M;RM\} \{0,1;0,3;0,5\}$  , где

$\{2,3\}$  – кол-во родителей, участвующих в операторе кроссовера;

$\{R;M;RM\}$  – оператор мутации (Reverse, Move, Reverse+Move соответственно);

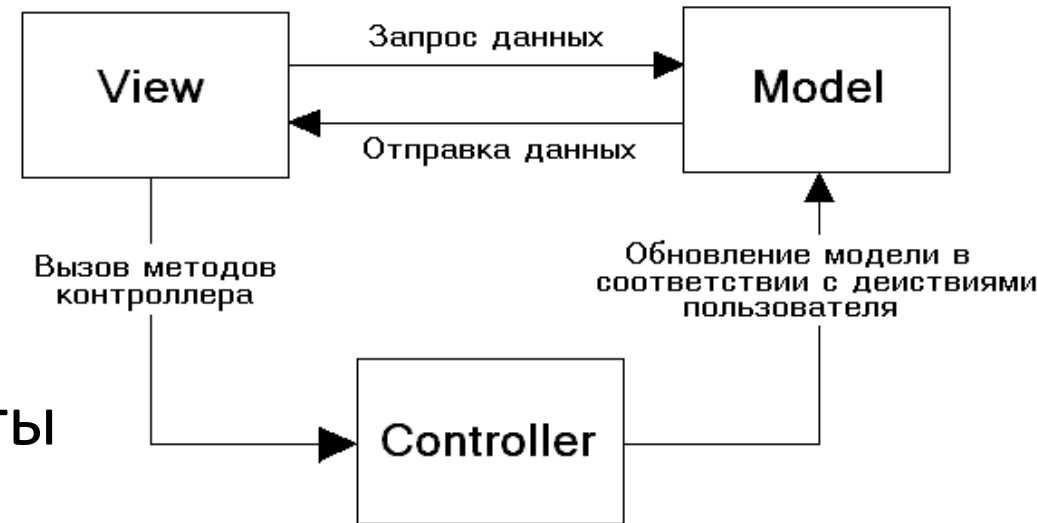
$\{0,1;0,3;0,5\}$  – значение вероятности мутации.

Например, 3R0,5.

# Структура ПО

- Язык программирования – Java 8
- Spring Framework (Web MVC, Data, Security)
- СУБД – MySQL
- ORM – Hibernate

- Картографический сервис – Яндекс.Карты
- Геокодирование – Яндекс Геокодер
- Получение расстояний между геообъектами - Mapbox Matrix Service





# Методика проведения вычислительного эксперимента (1 этап)

## 1 этап - На основе существующих решенных задач коммивояжера

- I. В качестве исходных данных взяты файлы из проекта TSPLIB, который предоставляет частные случаи задач коммивояжера разной размерности и их **оптимальные** решения.
- II. Написан модуль программы, считывающий исходные данные из файла и проверяющий решение на оптимальность.
- III. Каждая задача решается 10 раз каждой из модификаций генетического алгоритма.
- IV. Полученные значения целевой функции для каждых 10-ти повторных решений усредняются, сравниваются с оптимальным значением и вычисляется точность решения по формуле

$$T = \frac{F - OPT}{F} * 100\%,$$

где T – точность в процентах,

F – усредненное значение целевой функции алгоритма,

OPT – оптимальное решение задачи.

- V. На основе полученных результатов делается предположение о точности модификаций алгоритма.

# Результаты вычислительного эксперимента (1 этап)

Кол-во городов / Алгоритм	17	26	43	53	70	100
2R0,1	99,5%	91,0%	99,5%	77,7%	89,7%	71,8%
2M0,1	99,7%	92,0%	99,6%	78,7%	89,9%	73,2%
2RM0,1	100,0%	93,9%	99,5%	78,9%	89,8%	72,5%
3R0,1	100,0%	96,7%	99,7%	80,2%	90,3%	74,8%
3M0,1	100,0%	96,1%	99,5%	79,2%	90,5%	74,7%
3RM0,1	100,0%	96,2%	99,6%	79,8%	90,3%	73,5%
2R0,3	99,7%	94,8%	99,6%	78,4%	89,9%	72,1%
2M0,3	99,2%	94,3%	99,6%	78,3%	89,8%	71,6%
2RM0,3	100,0%	93,7%	99,5%	78,9%	89,8%	70,9%
3R0,3	100,0%	95,9%	99,7%	80,0%	90,3%	74,2%
3M0,3	100,0%	96,2%	99,7%	80,1%	90,3%	74,8%
3RM0,3	100,0%	95,2%	99,7%	80,1%	90,6%	74,9%
2R0,5	100,0%	93,8%	99,6%	78,1%	89,7%	72,9%
2M0,5	99,2%	94,0%	99,6%	78,6%	89,9%	70,9%
2RM0,5	99,7%	93,7%	99,5%	78,5%	89,7%	71,9%
3R0,5	100,0%	94,9%	99,8%	80,2%	90,6%	74,4%
3M0,5	100,0%	95,4%	99,7%	80,1%	90,5%	74,4%
3RM0,5	100,0%	97,0%	99,7%	79,7%	90,4%	74,3%

**Красным** выделены максимальные значения в каждом столбце.

# Результаты вычислительного эксперимента (1 этап)



**Вывод:** точность алгоритма не зависит от количества городов в условии задачи и варьируется в диапазоне [70%, 100%] Также заметно, что точность алгоритма меняется незначительно в зависимости от модификации алгоритма. Наиболее точными алгоритмами в целом являются 3R0,5 и 3RM0,3.

# Методика проведения вычислительного эксперимента (2 этап)

## 2 этап - На основе реальных данных

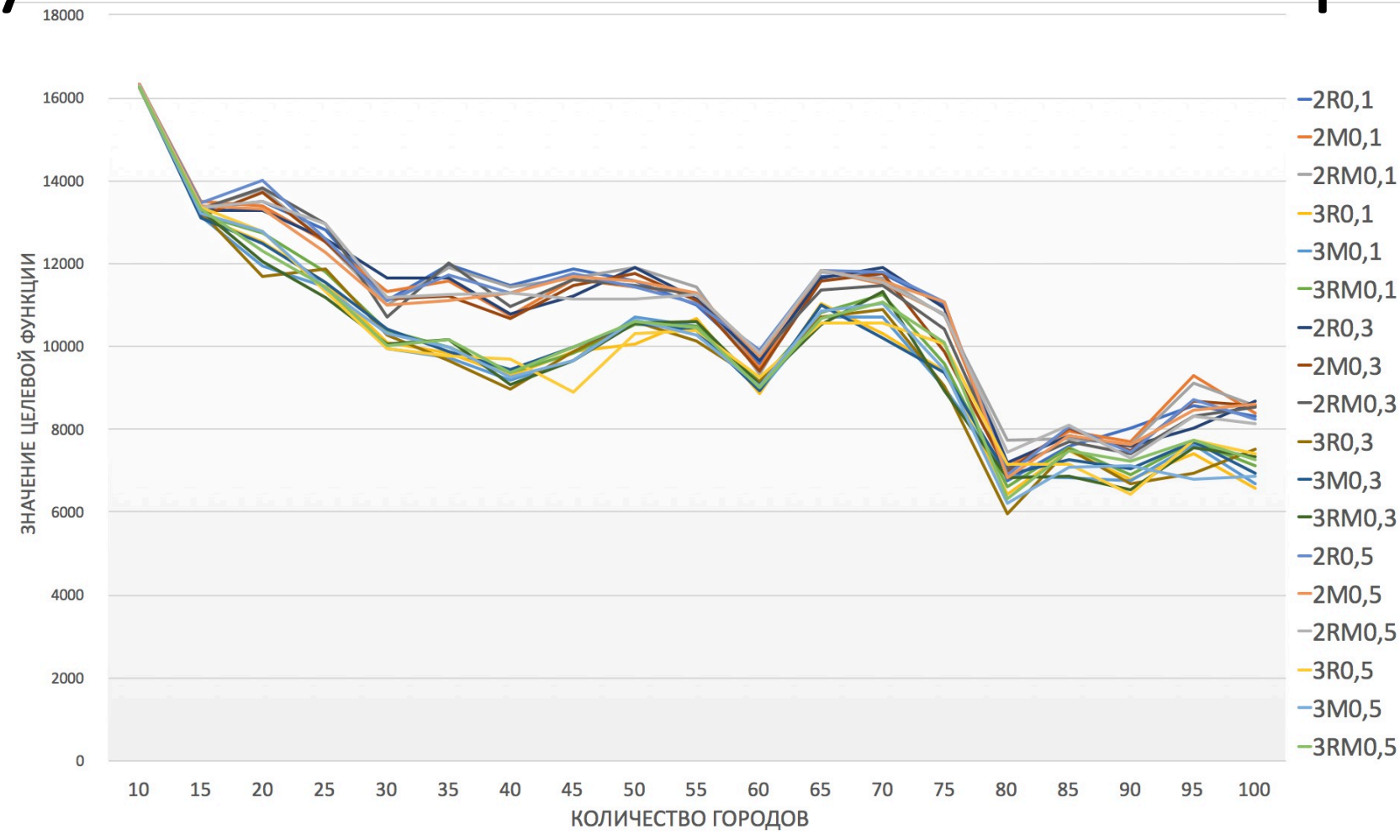
- I. В качестве исходных адресов для проведения эксперимента были взяты адреса г. Уфы и Уфимского района из Федеральной информационной адресной системы (ФИАС);
- II. Написан модуль программы, генерирующий случайную выборку от 10 до 100 адресов.
- III. Для каждой выборки решается задача построения рационального маршрута генетическим алгоритмом 10 раз каждой из модификаций генетического алгоритма.
- IV. Полученные значения целевой функции для каждых 10-ти повторных решений усредняются и сравниваются между собой.

# Результаты вычислительного эксперимента

Кол-во городов / Алгоритм	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80	85	90	95	100
2R0,1	16288,37	13337,06	13511,38	12805,95	11106,65	11976,46	11474,98	11866,2	11596,23	11006,02	9571,13	11671,28	11755,86	10999,4	6749,31	7582,45	8036,99	8584,2	8329,11
2M0,1	16327,47	13512,01	13407,66	12532,26	11325,17	11589,47	10733,51	11601,54	11428,63	11298,14	9477,86	11817,8	11512,63	10789,14	7072,43	7938,49	7697,05	9287,51	8381,54
2RM0,1	16310,43	13327,45	13786,76	12536,04	10985,63	11919,53	11429,15	11664,94	11913,94	11424,98	9321,91	11797,14	11661,38	10738,01	7721,8	7777,81	7641,66	9096,35	8560,39
3R0,1	16257,8	13125,42	12533,93	11536,12	10107,82	9835,68	9279,39	9889,23	10070,07	10684,38	8848,37	11034,3	10322,95	9408,05	6439,38	7472,97	6805,11	7403,17	6577,73
3M0,1	16258,17	13150,09	11924,3	11432,32	9963,53	9715,26	9172,33	9662,51	10700,36	10483,49	8944,75	10721,43	10725,3	9010,4	6877,56	6827,94	6762,89	7618,22	6698,64
3RM0,1	16261,43	13192,19	12728,76	11783,9	10396,2	9990,2	9381,44	9825,91	10510,41	10494,43	9158,77	10806,11	11240,25	9584,62	6612,63	7571,84	6891,63	7691,24	7106,64
2R0,3	16293,13	13272,78	13274,73	12595,72	11663,24	11636,73	10779,96	11226,15	11904,21	11134,56	9644,94	11651,89	11920,33	10917,61	7191,33	7841,45	7595,06	8034,81	8673,25
2M0,3	16297,13	13128,32	13736,18	12525,33	11144,39	11231,91	10664,74	11480,61	11745,05	11113,51	9417,85	11590,68	11743,22	9879,93	6991,83	8035,24	7482,88	8661,11	8571,51
2RM0,3	16323,47	13319,96	13815,18	12967,04	10712,26	12031,67	10978,14	11608,25	11482,16	11245,38	9757,51	11376,36	11488,17	10408,61	7030,81	7715,26	7400,61	8315,68	8528,21
3R0,3	16257,8	13237,46	11696,32	11882,41	10265,64	9644,71	8980,29	9887,14	10598,85	10126,58	9151,64	10710,5	10893,55	9039,53	5959,39	7519,49	6689,75	6934,27	7519,38
3M0,3	16276,23	13098,8	12491,5	11532,08	10417,33	9871,75	9435,2	9978,65	10567,21	10381,33	8917,21	10999,56	10219,49	9360,57	6916,11	7267,2	7065,23	7706,3	6930,49
3RM0,3	16265,06	13267,36	12051,29	11164,78	10065,08	10167,38	9084,48	9655,44	10570,64	10602,62	9066,88	10532,91	11328,16	8927,8	6828,71	6883,15	6549,26	7566,65	7345,48
2R0,5	16340,09	13449,48	13995,5	12600,15	11108,04	11729,55	11295,13	11745,32	11426,08	11022,5	9894,48	11824,14	11783,12	11085,22	6899,18	8064,08	7459,89	8731,33	8240,37
2M0,5	16324,07	13380,81	13321,4	12285,23	11002,67	11103	11293,68	11676,94	11577,07	11277,2	9765,76	11830,3	11583,49	11089,8	6817,4	7846,81	7610,42	8474,65	8606,54
2RM0,5	16297,5	13355,87	13518,77	12974,25	11180,01	11239,66	11304,8	11145,9	11129,8	11242,8	9826,37	11838,97	11546,67	10779,59	7435,5	8090,66	7306,85	8321,5	8118,02
3R0,5	16257,8	13360,25	12787,89	11341,39	9960,42	9759,45	9676,62	8896,98	10320,36	10373,11	9271,55	10562,72	10581,46	10059,74	7168,72	7143,64	6444,32	7748,74	7396,86
3M0,5	16272,6	13211,3	12789,63	11388,6	10305,85	10000,42	9250,25	9672,5	10627	10278,17	9049,56	10845,35	11021,63	9443,47	6199,05	7067,5	7119,73	6775,97	6870,02
3RM0,5	16265,06	13334,05	12301,9	11442,04	10012,28	10160,79	9367,95	9982,09	10615,2	10449,5	8996,61	10673,29	11075,14	10077,26	6338,78	7481,95	7219,99	7731,62	7271,47

**Красным** выделены минимальные значения в каждом столбце.

# Результаты вычислительного эксперимента



**Вывод:** на основе полученных результатов можно предположить, что оператор кроссовера с 3-мя родителями лучше справляется с решением задачи. Среди используемых операторов мутации лучше всего проявил себя REVERSE. В большинстве случаев лучшее решение находилось при вероятности мутации = 0,3.

**Поэтому для разработки приложения использовался алгоритм 3R0,3.**

# Интерфейс программы

Регистрация заявок в системе

Вход в систему и регистрация водителей в системе

Добро пожаловать, admin!

Добро пожаловать

Пункт производства

Водители

Заказы

Создать маршрут

Выйти

Добавить заказ

Полный адрес (город, улица, дом)

Дата доставки заказа: 20.06.2017

Описание заказа

Добавить заказ

Заказы :

№	Дата	Адрес	Статус	Описание	Отменить заказ
1	2017-06-20	Россия, Республика Башкортостан, Уфа, улица Карла Маркса, 59	Не выполнен	-	Отменить
2	2017-06-20	Россия, Республика Башкортостан, Уфа, улица Пушкина, 79	Не выполнен	-	Отменить
3	2017-06-20	Россия, Республика Башкортостан, Уфа, улица Шота Руставели, 25/1	Не выполнен		Отменить
4	2017-06-20	Россия, Республика Башкортостан, Уфа, улица Космонавтов, 4	Не выполнен		Отменить
5	2017-06-20	Россия, Республика Башкортостан, Уфа, улица Софьи Перовской, 44	Не выполнен		Отменить
6	2017-06-20	Россия, Республика Башкортостан, Уфа, улица Менделеева, 177	Не выполнен		Отменить
7	2017-06-20	Россия, Республика Башкортостан, Уфа, проспект Октября, 22	Не выполнен	-	Отменить
8	2017-06-20	Россия, Республика Башкортостан, Уфа, улица Рихарда Зорге, 70	Не выполнен		Отменить
9	2017-06-20	Россия, Республика Башкортостан, Уфа, улица Кирова, 44	Не выполнен	-	Отменить

Вход

admin

.....

Войти

Создание аккаунта

Хаматханов Ирек Винерович

irek

....

....

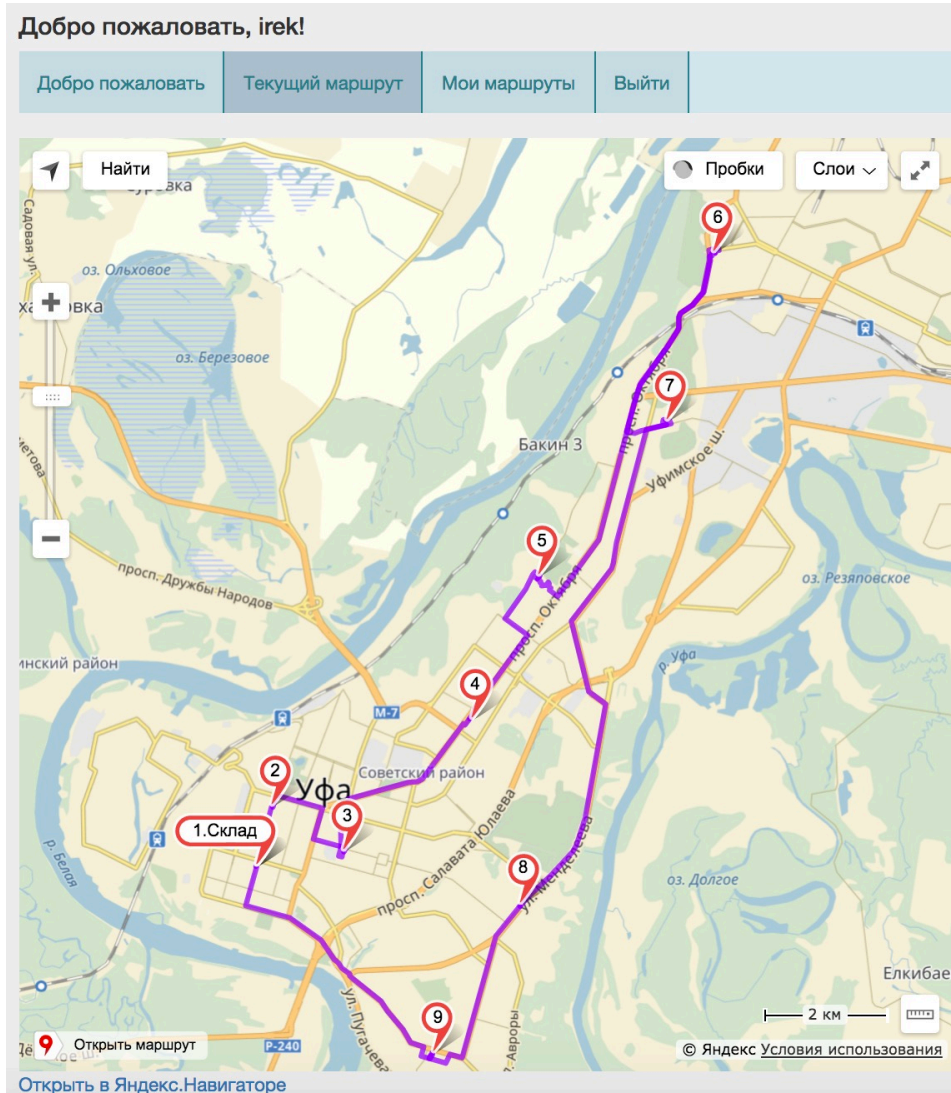
Россия, Республика Башкортостан, Уфа, Российская улица, 25 ↕

Создать



# Интерфейс программы

## Построенный маршрут в личном кабинете водителя



Трогаемся,

Едем прямо на улица Карла Маркса, проезжаем 1041.17 м.,  
Едем прямо на улица Карла Маркса, проезжаем 208.32 м.,  
Едем направо на Революционная улица, проезжаем 798.28 м.,  
Едем направо по двору, проезжаем 540.67 м.,  
Едем налево на улица Кирова, проезжаем 559.43 м.,  
Едем направо по двору, проезжаем 50.84 м.,  
Едем направо по двору, проезжаем 33.05 м.,  
Едем прямо по двору, проезжаем 69.86 м.,  
Едем направо по двору, проезжаем 53.49 м.,  
Едем направо на улица Пархоменко, проезжаем 961.53 м.,  
Едем направо на улица 50-летия Октября, проезжаем 1308.24 м.,  
Едем прямо на проспект Октября, проезжаем 1094.48 м.,  
Едем направо по двору, проезжаем 32.61 м.,  
Едем налево по двору, проезжаем 128.65 м.,  
Едем прямо по двору, проезжаем 183.46 м.,  
Едем налево по двору, проезжаем 32.18 м.,  
Едем направо на проспект Октября, проезжаем 1502.72 м.,  
Едем налево на улица Шафиева, проезжаем 468.26 м.,  
Едем направо на улица Рихарда Зорге, проезжаем 480.77 м.,  
Едем левее на улица Рихарда Зорге, проезжаем 443.12 м.,



# Выводы

- Проведен обзор существующего ПО для поиска оптимальных маршрутов. Выяснено, что не существует бесплатных программ, которые строят и визуализируют на карте рациональные маршруты для большого количества городов.
- Выполнен аналитический обзор методов решения задачи. На основе анализа выбран генетический алгоритм в силу его большой гибкости при настройке параметров.
- Разработаны и программно реализованы различные модификации генетического алгоритма.
- Произведен вычислительный эксперимент, который показал:
  - Точность решения задачи не зависит от количества городов;
  - В операторе кроссовера использование трех родительских особей улучшает результат;
  - В качестве оператора мутации лучше всего себя проявил оператор разворота последовательности генов.
- Написано программное обеспечение для построения рационального маршрута и визуализации его на карте.
- Проведен расчет на реальных данных.

Спасибо за внимание!