**Министерство науки и высшего образования Российской Федерации**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«Национальный исследовательский университет ИТМО»**

**(Университет ИТМО)**

**Факультет Институт прикладных компьютерных наук**

**Образовательная программа 01.04.02**

Отчет

по лабораторной работе №4

по дисциплине **«Генетические алгоритмы»**

Выполнил: **студент группы M4130 Горбатовский А. В.**

Проверил: **Муратов С. Ю.**

Санкт-Петербург

2024

# Генетический алгоритм для задачи коммивояжёра

## Цель работы.

Получить навыки разработки для комбинаторных задач на примере задачи коммивояжёра.

## Ход работы.

### Класс TspSolution служит для представления индивидуальных решений задачи коммивояжёра, обертывая в себе массив целочисленных значений, которые представляют индексы городов в порядке их посещения. Этот подход не только обеспечивает гибкость в генерации и модификации маршрутов через случайную перестановку или заданную последовательность, но также предлагает базовые операции над маршрутами, такие как получение, установка значений и обмен элементов, что является ключевым для реализации генетических операторов мутации и кроссовера.

### В процессе настройки фитнес-функции используется класс TspFitnessFunction, который зависит от объекта TspProblem, содержащего информацию о вершинах графа и их координатах. Фитнес-функция вычисляется путем последовательного обхода маршрута, заданного в TspSolution, и подсчета суммарного евклидова расстояния между последовательными вершинами. Это обеспечивает количественную оценку маршрута, где меньшие значения фитнес-функции указывают на более короткие и, следовательно, более предпочтительные маршруты. Фабрика кандидатов инициирует создание экземпляра с использованием конструктора для генерации кандидата случайным образом.

### Мутация определяется двумя числовыми генераторами:

### Генератор для определения числа мутаций.

### Генератор для определения дистанции между позициями элементов, которые будут переставлены.

### В процессе мутации используются гауссовские семплеры, что позволяет моделировать количество мутаций и дистанцию между переставляемыми позициями в соответствии с распределением Пуассона. Это обеспечивает более естественное распределение изменений, имитируя случайные мутации с различной интенсивностью. Такой подход способствует более тонкой настройке генетического алгоритма, позволяя адаптироваться к разнообразным ландшафтам поиска и улучшая способность алгоритма исследовать пространство решений.

### В процессе мутации индивида первоначально определяется количество перестановок с помощью сэмплирования, после чего указанное число раз производится выбор пары элементов массива для их последующей перестановки местами.

### Для кроссовера был выбран метод сохранения средней части. Этот подход заключается в выборе левой и правой границы части, которая будет сохранена из одного из родителей в потомка. Затем, перебирая элементы, оставшиеся от второго родителя, добавляем их в потомка, сохраняя исходный порядок. Границы для сохранения части выбираются с использованием равномерного распределения.

### Были исследованы различные стратегии выбора для оптимизации процесса эволюции, включая RouletteWheelSelection, TournamentSelection с вероятностью 0.8, и RankSelection. Окончательный выбор был сделан в пользу RankSelection, которая оказалась наиболее эффективной в данной реализации задачи коммивояжёра.

### Результаты тестирование представлены в таблице 1.

Таблица 1 – Результаты экспериментов

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Проблема | Размер | Параметры popsize и gens | Длина маршрута | Количество итераций до сходимости | Оптимальный маршрут |
| XQF131 | 131 | 100; 100000 | 698.13 | 98694.8 | 564 |
| XQG237 | 237 | 100; 100000 | 1927.17 | 99541.2 | 1019 |
| PMA343 | 343 | 100; 100000 | 3147.22 | 99903.8 | 1368 |
| PKA379 | 379 | 100; 100000 | 3342.8 | 99403.5 | 1332 |
| BCL380 | 380 | 100; 100000 | 4675.98 | 99583.2 | 1621 |
| PBL395 | 395 | 100; 100000 | 3875.28 | 99213 | 1281 |
| PBK411 | 411 | 100; 100000 | 4071.52 | 99487.8 | 1343 |
| PBM436 | 436 | 100; 100000 | 4641.6 | 99315 | 1443 |

## Ответы на вопросы:

### Определение глобального оптимума для полученного решения возможно через точное решение TSP, что в случае экспоненциальной сложности задачи не является практичным подходом. Так как TSP относится к классу NP-трудных задач, особенно при условии ограничения по длине пути, точное решение недостижимо за приемлемое время. Альтернативный метод с использованием недетерминированной машины для решения TSP с уменьшенным на единицу ограничением длины пути может указать на глобальный оптимум, если более короткого пути не существует. Однако такой метод не является реализуемым в рамках алгоритмов, работающих за разумное время.

### Принимать решения с повторениями городов можно, но поскольку они не соответствуют корректному определению решения TSP, их необходимо отсеивать в процессе вычисления фитнес-функции. Наличие невалидных решений может замедлить алгоритм из-за необходимости проведения дополнительной валидации для каждого кандидата в популяции. В худшем случае, такие решения могут доминировать в популяции, что снизит вероятность генерации валидных кандидатов и снизит эффективность алгоритма.

### Изменение задачи путем устранения требования возврата к исходной точке маршрута ведет к необходимости модификации фитнес-функции: следует исключить добавление дистанции замыкающего ребра. Оставить текущую методику генерации и представления решений можно без изменений, поскольку она по-прежнему формирует допустимое решение. Вместе с тем, следует учесть, что множество потенциальных решений увеличивается в количество раз, равное числу вершин в графе, поскольку каждое циклическое решение теперь приводит к множеству путевых решений при удалении одного ребра. Это увеличение пространства поиска может привести к замедлению работы алгоритма.

Ссылка на репозиторий: [github.com/Myashka/ITMO\_Genetic\_algorithms](https://github.com/Myashka/ITMO_Genetic_algorithms)