

# Решение задачи TSP

Александр Ващилко, 874

May 18, 2020

## Introduction

Задача Euclidian TSP нам была задана в рамках курса Дискретной оптимизации, поэтому здесь я опишу используемый мной подход к решению этой задачи. Всего в рамках решения задачи я использовал: метод ближайшего соседа, спрямления, а также последующее улучшение этих решений с помощью local search с поиском в 2- и 3- окрестностях.

## Ближайший сосед

Метод ближайшего соседа предполагает выбор начальной точки обхода, а далее жадным образом добавляем в путь точку, ближайшую к последней точке текущего пути.

## Спрямление

Для начала нам необходимо получить начальный цикл, проходящий через все вершины графа (не обязательно один раз). Для этого построим MST с помощью алгоритма Краскала или Прима (я использовал Краскала), а затем с помощью обхода в глубину построим замкнутый путь, проходящий через все вершины графа. Для этого модернизируем DFS: будем помещать в массив номера вершин при входе в вершину и при выходе из нее. Обходим полученный граф-цикл, конструируя новый граф, и на каждом шаге делаем следующее: если вершины нет в новом графе добавляем ее в новый граф, иначе идем дальше. Полученный граф является гамильтоновым.

## Local search

Локальный поиск предполагает нахождение в некоторой окрестности текущего решения  $x$  более эффективного решения  $y$ . Тут возможно несколько подходов: остановиться, как только нашел более эффективное решение или продолжить поиск с целью нахождения наиболее эффективного решения. Затем,  $x = y$ , и снова запускаем итерацию. Число итераций контролируется гиперпараметрами, например непосредственно числом итераций или минимальным отношением прироста к текущему значению оптимума. Также существует модификация local search, допускающая выбор менее эффективного решения с целью возможности выхода из локальных оптимумов.

## Что использовалось на практике

На практике были наложены ограничения по времени, поэтому улучшению, позволяющее выйти из локальным минимумов не было сделано. Я предложил два варианта решения: спрямление с запуском DFS из разных начальных точек, метод ближайшего соседа с последующим локальным поиском в 3-окрестности. Также стоит отметить, что минимально решение ближайшего соседа отнюдь не приводит к минимальности решения после локального поиска, поэтому стоит минимизировать решение, полученное после локального поиска.