Команда “**Hackathon runners**” задача №6.

Здравствуйте, мы хотим представить описание нашего решения и способ его запуска на вашей системе.

Мы реализовали решение вашей задачи на языке Python в формате тетрадки “.IPYNB”. Принцип работы нашего алгоритма вкратце:

Нами были проведены тесты алгоритма поиска путей, где мы искали лишь пути по ближайшим точкам. Такой метод позволил достичь количества точек около 55 для областей высокой концентрации терминалов и количества точек около 30 для зон низкой концентрации. 3 машины будут иметь в идеальном случае 150 - 160 точек в маршруте, чего, кажется, должно хватать. Однако на практике точки заполняются совсем неоднородно, пути приходится строить по всей карте москвы и мы считаем что для 3-х машин с заданными условиями (посещение хотя бы раз в 14 дней и посещение при накоплении миллиона) неразрешима. А так как, по нашему анализу, основные расходы связаны именно с содержанием машин, уменьшение количество автомобилей в парке стоит в приоритете. Поэтому перед нами стояла задача решения задачи для 4-х машин, с чем мы и справились. Оптимизация по остальным статьям расходов также учитывалась.

Наше решение можно разбить на несколько частей.

Самая важная часть нашего решения - создание критерия приоритетности посещения терминалов, столбец "points" матрицы MoneyAmount (также содержится в матрице coords для построения графиков).

Данная оценка позволяет следующей части нашего алгоритма - функции построения маршрутов, стабильно решать задачу выполнения условий для каждого дня, при этом упрощая решения для последующих дней.

Благодаря нашей системе оценки карта терминалов делится на “зоны плотности”, созданные путями маршрутов, что помогает их построению в будущем.

В этой системе учитывается заполненность терминала деньгами и время его последнего посещения, а также, для тренировочных данных найденное среднее поступление денег в банкомат, а для тестовых - спрогнозированные данные n-beats модели на 30 дней, полученные обучением на первых 60 днях. Данные были выгружены программой "Forecast\_time\_series\_model.ipynb" в файл "forecast\_last\_30\_days\_data.xlsx" и загружены в этом файле перед основной функцией. Использование спрогнозированных данных позволяет уточнять оценки, а значит стабилизировать весь алгоритм.

Система очень гибкая, параметров много, и все можно настраивать, поэтому она может быть применима и в других задачах логистики.

Настройка параметров, которые установлены в ней сейчас, велась посредством тестов и сравнения целевых показателей (расходов, срыва условий). В результате мы добились соблюдения условий для 4-х машин по тестовому месяцу, и общих затрат по нему около 3.530.000 рублей.

Стоит заметить, что после построения путей происходит их оптимизация по времени алгоритмом perturbation algorithm по схемам two optimum либо ps6 (настраивается в ручную). Данные схемы используют стохастические методы, а значит результаты могут незначительно отличаться. Из этого следует, что маршруты могут строить по разному на долгих промежутках времени. Однако, стоит заметить, система стабильна - при каждом запуске нарушений условий не было.

В комментариях в коде описан принцип работы каждой из функций.