

Magenta Contest 2019

Описание задачи:

Одним из основных направлений работы в Magenta Technology является построение алгоритмов для автоматического решения различных задач транспортной логистики. Примером такой задачи, часто встречающейся на практике, является широко известная задача коммивояжера.

Формально решение задачи коммивояжера заключается в нахождении цикла минимальной длины, проходящего через все вершины ориентированного графа ровно один раз. В реальной жизни нахождение решения этой задачи отвечает на вопрос о том, в каком порядке курьер должен объехать своих клиентов, чтобы минимизировать транспортные расходы.

На практике клиенты обычно имеют ограниченное окно доступности, которое курьер должен учитывать при построении маршрута. Поддержка такого ограничения является обязательным требованием для решения реалистичных задач. В рамках данного конкурса мы предлагаем попробовать свои силы в решении задач такого рода.

Рассмотрим рабочий день интернет-магазина. В начале рабочего дня курьер получает на складе товары, которые ему необходимо доставить в течение дня. Для каждой посылки известен адрес доставки и временное окно, в течение которого клиент будет ждать на адресе доставки. Задача курьера - доставить товары всем клиентам в его списке и вернуться на склад как можно раньше. Если курьер не успевает выполнить доставку во временное окно доступности клиента, или не выполняет доставку вообще, то клиент отказывается от доставки и компания теряет деньги. Если он приезжает раньше, чем клиент станет доступен, то он должен подождать клиента.

Вашей задачей будет помочь курьеру определить оптимальный порядок доставки заказов. Из двух вариантов решения задачи более оптимальным будет считаться тот, в котором будет меньше число отказов от доставки из-за опоздания курьера. При одинаковом числе отказов более оптимальным будет считаться вариант решения, в котором возвращение на склад происходит раньше.

Целевая функция:

Для оценки стоимости решения конкретной задачи используется следующая функция:
 $Стоимость = \langle \text{Время возвращения в депо} \rangle + \langle \text{Число отказов от доставки} \rangle * \langle \text{Штраф за отказ от доставки} \rangle,$

где

$\langle \text{Штраф за отказ от доставки} \rangle = \langle \text{максимальная длительность переезда в матрице длительностей} \rangle * \langle \text{число заказов в задаче} + 1 \rangle.$

Штраф за отказ от доставки задан таким образом, чтобы получение отказа всегда было менее выгодно, чем любая оптимизация пути. Таким образом, расписание без отказов от доставки всегда более выгодно, чем расписание с отказами.

Модель:

Граф, для которого необходимо найти решение, полносвязный, т.е. существует прямой маршрут между каждым объектом доставки и всеми другими объектами доставки и складом. Граф, на котором необходимо найти оптимальное решение, представлен описанием вершин и матрицей смежности, ячейки которой содержат время, необходимое для перемещения между этими вершинами.

Маршрут курьера начинается в момент времени 0 (посещение склада). Затем курьер перемещается в точку обслуживания первого заказа в маршруте, затрачивая на это время, записанное в соответствующей ячейке матрицы. Если курьер приезжает раньше времени доступности заказа, то курьер дожидается времени доступности и совершает разгрузку. Если курьер приезжает позже времени доступности заказа, то клиент оформляет отказ от доставки. На точке доставки курьер мгновенно разгружается (время разгрузки считается нулевым) и отправляется к следующему заказу в маршруте. После разгрузки на последней точке доставки курьер перемещается на склад. По возвращении на склад рабочий день завершается и курьер получает оплату согласно числу доставленных в срок посылок и затраченному времени. В случае, если курьер не посещает точку заказа в рамках маршрута вообще, то считается что клиент оформляет отказ от доставки.

Формат входного файла:

Во первой строке записано целое число N ($1 \leq N \leq 100$) - количество заказов, которые курьер должен доставить в течение дня.

Во второй строке через пробел записаны координаты x_d и y_d склада.

В каждой из следующих N строк через пробел записаны координаты x_i , y_i и начало и конец временного окна s_i , e_i заказа, который необходимо доставить. Для всех заказов времени гарантируется, что $s_i < e_i$.

В следующих $N+1$ строках записана полная асимметричная матрица длительностей переездов между точками. В ячейке $[j,k]$ матрицы содержится время переезда от точки j до точки k , где $0 \leq j, k \leq N$. (Считаем что склад имеет индекс 0).

Все координаты заданы целыми числами, не превосходящими по модулю 10^5 .

Моменты времени и длительности представлены целыми числами в диапазоне $[0; 10^5]$.

Формат выходного файла:

В единственной строке необходимо через пробел вывести список номеров заказов, соответствующий оптимальному порядку объезда.

Ограничения:

На нахождение решения одной задачи алгоритму отводится не более 10 секунд с учетом ввода/вывода данных.

Пример:

Входной файл	Выходной файл
3 0 0 0 2 0 20 2 2 10 30 2 0 20 30 0 10 15 10 10 0 10 15 15 10 0 10 10 15 10 0	1 2 3

Наборы данных:

Набор тестовых данных разделен на 2 части - отладочная и проверочная выборка. Отладочная выборка получена выделением части задач из проверочной выборки и доступна публично для тестирования и отладки алгоритмов. Отладочная выборка может быть расширена после начала конкурса, о чем будет сообщено дополнительно. Проверочная выборка будет использована при финальном ранжировании стратегий и будет опубликована вместе с результатами конкурса.

Оценка решений:

После запуска алгоритмов на проверочной выборке, каждому из них будет выставлена оценка, и решения будут отсортированы по убыванию этой оценки. Победителем конкурса будет считаться алгоритм, показывающий наилучшие результаты на большем числе задач по сравнению с другими участниками.

Для каждой из задач будет вычислено значение относительного качества планирования:

Качество решения задачи = <Лучшая стоимость решения задачи среди всех участников> / <Стоимость решения задачи>.

Если на конкретной задаче стратегия упадет или не уложится во временной лимит, то качество решения этой задачи будет считаться равным нулю.

Сумма полученных значений на всех задачах с весовыми коэффициентами (которые будут зависеть от размерности и сложности задачи) будет определять рейтинг стратегии, по которому будет выполнено ранжирование. Стратегия с наивысшим рейтингом будет считаться победившей.

Отправка решения:

Для выполнения решений допускается использовать один из следующих языков программирования: C/C++/Java/C#/Python/Pascal. По просьбам и при содействии участников список языков программирования может быть расширен. Допускается использование только стандартных средств выбранного языка программирования.

Ожидается, что программа принимает на вход файл “salesman.in” и выводит необходимую последовательность объезда заказов в файл “salesman.out”.

Решения необходимо присылать на адрес электронной почты contest@magenta-technology.com. В теме письма необходимо указать [TSP Contest 2019] и имя участника. Допускается неограниченное число посылок решений, но в финальном тестировании будет участвовать последняя отправленная. Аналогично данный адрес можно использовать для возникающих вопросов по задаче.

Решения принимаются до 11 апреля включительно.

По мере поступления решений будет вручну поддерживаться таблица с текущими результатами по следующей ссылке: <http://tiny.cc/5d8o4y>

Участие в конкурсе индивидуальное. Призы будут вручаться участникам Чемпионата Поволжья в двух категориях: приз за решение с наилучшим качеством и приз симпатий жюри за наиболее оригинальный алгоритм решения задачи.