***Цель: Снижение размерности исходных данных в задаче маршрутизации***

В файле «Несколько дней» содержатся все исходные данные по заказам за три дня. Будем рассматривать только какой-нибудь один день.  
Допустим, имеем дело с компанией «Чистая вода» или «Норинга», это неважно. Рабочий день начинается в 8 часов и заканчивается в 20.  
В компанию поступают заказы в разные точки города (известны их координаты) : привезти воду, кулер, либо приехать заключать договор (в данном случае в поле веса будет 0). Нам не важно, что доставлять, важен только вес.  
Соответственно, посылаются машины со склада (его координаты известны), грузоподъёмность которых 2500 кг. И конечно, чем меньше машин будет задействовано, тем лучше. Скорость движения пусть будет постоянной и равной 50 км/ч.  
Клиент говорит желаемую временную рамку приезда машины, например, с 13 до 15 часов. То есть в это время машина должна приехать и успеть обслужить, заранее известно время обслуживания каждой точки.  
Суть задания в том, чтобы минимизировать количество точек путём агрегации (объединения в одну).

То есть в случае, если точки находятся очень близко друг к другу, скажем, в шаговой доступности (это может быть, например, соседний подъезд или соседний дом, тут надо самому задать ограничение, например, 500 метров попробовать) , а также их временные рамки пересекаются и общий вес заказов, которые везет машина, не превышает 2500 кг, тогда мы эти точки агрегируем в одну.  
Пример:  
Две точки находятся близко:  
1) с 13 до 15 часов, обслуживание 5 мин, 5 кг.  
2) с 11 до 14 часов, обслуживание 6 мин, 1 кг.  
У новой точки будет: общая временная рамка с 13 до 14; сначала обслужим ту точку, которая ближе; общий вес 6 кг; время плюсуется примерно так = время сейчас + время доехать до ближайшей точки + время её обслуживания + доехать до второй точки + время её обслуживания. Мы должны уложиться во временную рамку, то есть, чтобы не было случаев, что мы не успеваем обслужить клиента в поставленные временные рамки. В агрегированную точку может входить сколько угодно точек, тут на количество ограничения нет.  
Есть ограничение на общее время у агрегированной точки (сумма времени обслуживания точек и времени доезда из одной точки в другую) – его можно менять (по указаниям преподавателя значение этого ограничения пробовать ставить 1 час или 2 часа, но не более).  
Таким образом, мы должны сохранять данные по агрегированной точке (новый ID точки, усредненные координаты, общее время, вес и т.д.), в том числе данные точек, входящих в агрегированную.  
Кроме этого нужно вычислить количество агрегированных точек, сколько минимально необходимо машин, чтобы их обслужить, а также маршрут для каждой машины за день.  
Важно иметь ввиду, что могут быть «выбросы» - это точки, которые не удалось по каким-либо причинам объединить, но обслужить их всё равно нужно. Смотреть иллюстрацию ниже.  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
Входными данными является массив заказов с данными:  
OrderID - ID заказа,  
OrderWeight - вес заказа (кг),  
storeDistance - расстояние от склада (в м),

lat - широта,  
lon - долгота,

tStart – начало временной рамки,

tEnd – конец временной рамки,

tDuration – время обслуживания

\_\_\_\_

Примерный рисунок, но это в идеале.

Машина №2

Агрегированная точка

Агрегированная точка

склад

Машина №3

Машина №1

Агрегированная точка

Агрегированная точка

«Выброс»