车辆路径规划中的分配问题

背景介绍

在最近几年的移动互联网发展中，出现了许多为 解决用户**即时需求(On-Demand)** 的产业，尤其是在打车和外卖领域，它们都形成了很大的经济规模，产生了像 Uber、滴滴、美团、饿了么这样的大平台。

在整个市场中，平台要解决的一个核心问题是订单分配，比如在打车领域，用户会发起 1 个需求，可能是早上 8 点从小区出发到公司，然后平台会选择周围最合适的司机，把订单指派给他，让这个司机去接送用户。这种需求通常是随机的，并且有一定聚集性，可能在早上 8 点 10 分，一个小区周围会同时出现几十或上百个订单或需求。于是，如何快速地把这些订单分给周围的司机，既要满足最多的用户需求，又要使司机的服务成本最低（比如油费损耗、拥堵时间等），是一个很有挑战的问题。

问题描述

在业界，上述问题叫做**车辆路径规划问题 (Vehicle Routing Problem)**，除了前面提到的订单分配之外，它还包括另外一个路径规划的问题。同样拿打车举例，假设司机接了 2 个订单，他是先送完 1 个乘客再送另外 1 个乘客，还是先接 2 个乘客然后再分别送，不同的方案，花费的成本和时间都是不一样的。这其实就是**旅行商问题(Traveling Salesman Problem)**。这里只强调一点，因为路径规划方案的差异性，比如共乘或绕路，1 个司机 V1 接送 2 个用户 O1, O2 的总成本，通常不等于司机分别接送 2 个用户的成本之和，即成本是无法线性相加的 **Cost(V1, {O1, O2}) != Cost(V1, O1) + Cost(V1, O2)**。这是造成问题复杂度增加的一个主要原因。

本文后面只讨论订单分配的问题，即假设司机和订单之间的成本 Cost(V1, {O1, O2}), Cost(R1, O1), Cost(R2, O2) 都是已知的，问题的目标是找到一个分配方案，使得所有订单都能分配出去，且所有司机的成本之和最小。（和 1 个司机的情况不一样，不同司机之间的成本是可以累加。）

举 1 个例子，假设有 4 个司机 V1, V2, V3, V4，和 3 个订单 O1, O2, O3。它们之间的分配成本可以汇总在 1 个表格，如下，

表 1：分配成本表

	V1	V2	V3	V4
O1	2	3	3	X
O2	5	4	4	X
O3	3	4	**2**	5
O1,O2	5	**4**	4	X
O1,O3	5	6	5	X

其中，最优的分配方案是订单 O1, O2 分给 V2, 订单 O3 分给 V3，总成本是 TotalCost = Cost(V2, {O1,O2}) + Cost(V3, O3) = 4 + 2 = 6。下图是分配结果的示意图，

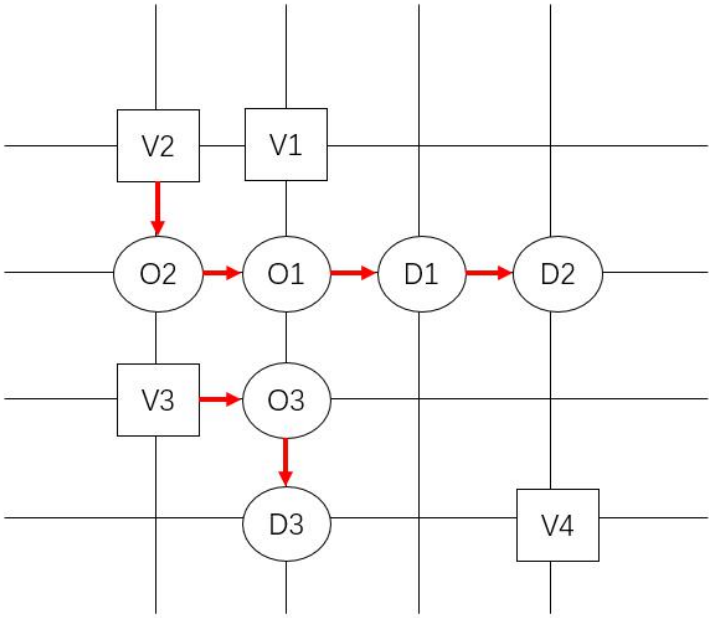

图 1：订单分配结果及路径规划

在这个例子之上，补充下面几个说明点。

1. 把 {O1,O2} 称做 1 个包，里面有 2 个订单。 实际当中也有 3 个订单以上的包 {O1,O2,O3}，但是在分配成本表中忽略掉了，包括{O2,O3}，因为这里认为把 O2,O3 打包到一起的效果不好。
2. 每个司机只能分配到 1 个包或 1 个订单，例如不能把 O3, {O1,O2} 同时分给 V1，因为之前讨论过，Cost(V1, {O1,O2,O3}) != Cost(V1, O3) + Cost(V1, {O1,O2}) 。类似地，每个订单也只能分配给 1 个司机。
3. 表 1 中，V4 对应的列元素有符号“X”，它表示这个订单或包不能指派给对应的司机，原因可能是接客里程过远等因素。
4. 图 1 表示简化的地图，V1,V2,V3,V4 分别表示每个司机的起始位置。O1, D1 分别表示乘客 1 的出发点 Origin 和目的地 Destination。其它的类似。
5. 在图 1 中，司机的成本简化成接送路线在网格中的长度。接送路线必须满足先接后送，如果有多条路线，取最短的 1 个。比如，Cost(V1, O1) = Dist(V1, O1, D1) = 2，Cost(V1, {O1,O2}) = Dist(V1, O2, O1, D1, D2) = 5。
6. 在分配结果中，V1, V4 没有分配到订单，所以它们的成本是 0，在总成本中不用考虑。司机比乘客多也是一种常见情况。

数据说明

测试数据会有 100 个任务，每个任务的第 1 行是任务信息，之后是具体的订单和司机的分配成本。比如，上面的例子会表示成

Task=1 Orders=3 Vehicles=4 Scores=16
1 1 1 2
1 1 2 3
1 1 3 3
1 2 1 5
1 2 2 4
1 2 3 4
1 3 1 3
1 3 2 4
1 3 3 2
1 3 4 5
1 1,2 1 5
1 1,2 2 4
1 1,2 3 4
1 1,3 1 5
1 1,3 2 6
1 1,3 3 5

分配成本的数据共有 4 列，分别是任务 ID，订单 ID(多个订单用逗号分割)，司机 ID，分配成本。另外有 2 个需要注意的地方，

1. 所有任务的数据都在一个文件中，需要根据任务 ID 区分。
2. **一个包里的订单数上限是 2。**即最多只有 2 个订单能拼到一起，暂不考虑 3 个订单及以上的情况。

结果评估

结果评估主要考虑 2 个指标。

1. **单均成本** AvgOrderCost=TotalCost / NumOrders。比如例子中的 AvgOrderCost = 6/3 = 2。在所有任务中，取得的最优值越多越好。
2. **求解耗时** SolverTime。最后会参考所有任务的平均时间，时间越短越好。

其它要求

1. 可以使用线性规划求解器，但不能使用整数规划求解器。

参考资料

1. Alonso-Mora J , Samaranayake S , Wallar A , et al. On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment[J]. Proceedings of the National Academy of Sciences, 2017, 114(3):462-467.