物流规划问题的一般模型和基于剪枝优化的全局最优解

摘 要

本文根据非线性离散问题的相关理论和 TSP 问题的处理流程，采用分治、剪枝优化等多种优化算法，给出了小规模物流规划问题的一般模型和全局最优解，使用主流个人电脑，算法可以在1 秒钟内完成计算。

我们给出的物流规划模型是多目标规划模型，主要将路程、客户对时间的要求作为规划的目标，将载重量等作为约束进行处理。

模型求解时，遵循严格遵循客户需求的原则，所以，客户对时间要求作为约束需要严格遵循。从而实现了化多目标为单目标。后续可以使用相对简单一些的单目标规划问题求解思路进行求解。

模型求解时，针对该类规划问题的特殊性，我们可以分步进行。首先计算单车运输的全部可行方案，经过剪枝处理后，只有 32 组可行方案。最后给出的多车全量覆盖客户的可行运输方案由这 32 组方案中的任意 n 组组合而成。问题规模大幅度缩小，从而实现了在短时间内计算出全局最优解。

主要创新点

1. 分步计算，先计算单车运输的全部可行方案，然后使用获得的 32 组方案排列组合即可获得全局最优解

关键词：单车运输的全部可行方案，多车全量覆盖客户的可行运输方案，全局最优解，分治算法，剪枝优化

目录

1 问题重述

某物流中心为若干个客户配送物资，物流中心与客户以及客户与客户之间的距离已知。送货车辆都从物流中心出发，最后回到物流中心，可以一次为多个客户送货，但最大载重量为有限值。尽可能在用户规定的时间范围内到达，否则成本将增加。

综合考虑所需车辆数、总运行里程和路径等因素，选取最佳配送方案。

2 问题分析

该问题与旅行商问题、中国邮递员问题等 TSP 问题接近，但有着根本区别：对每一个节点的到达时间有明确的要求。

TSP 问题本质上属于非线性离散的数学问题。当前各类求解非线性离散问题的方法，本质均是通过计算并比较所有可行解来给出最优解。在计算可行解的过程中，可以通过剪枝优化、分支界定、分治等算法缩小最优解所在的可行解的解空间。对于一些高度复杂的问题，因为算法时间复杂度问题，通常只能给出局部最优解。

本题中，所给数据规模较小，可以尝试尽可能的缩小计算范围，给出一个全局最优解。

2.1 关键影响因子分析

影响该问题决策的四个关键因素是：迟到时间、早到等待时间、总路程、总费用。

2.1.1 迟到时间

迟到通常伴随着高额的赔偿金、严重的信誉受损等间接影响。

从客户角度看，是否按时交付货物是衡量物流中心好坏的关键因素之一。

故，最终的解决方案中，不能存在迟到现象。

我们假设，在与客户达成的时间协议中，不存在无法满足的时间要求。

2.1.2 早到等待时间

与迟到不同，提前到达并不存在高额的赔偿金等问题。但是，商业活动中，守时依旧是极其重要的。在对方未准备好的时间提前将货物送到，可能会造成一些不可预估的负面影响。

在最终的解决方案中，我们优先选取守时的方案，但不排斥提前到达的方案。即使提前到达，也依旧要在客户规定的时间内完成交易（卸货等）。

2.1.3 路程

路程，是对成本影响最大的因素之一。在符合客观规律和客户要求等的基础之上，我们优先选取路程最短的方案。

2.1.4 全部时间消耗

全部时间消耗即从车辆离开物流中心道回到物流中心所需的时间，具体包括车辆运行时间和等待时间。

若存在提前到达现象，则会导致等待时间。等待时间也会在一定程度上增加成本，但相对于路程，等待时间对成本的影响较小。

为了减少等待时间也选择绕路等增加路程的方案，是不可取的。

我们可以通过计算全部时间消耗来有效规避该误区。

2.1.5 汽车载重量

载重量是必须遵守的客观因素。不存在例外的情况。

2.1.6 汽车数量

我们期望，汽车的数量越少越好。

2.2 算法的时间复杂度估计

根据排列组合和概率论的基本理论，我们可以知道：

复杂问题通常可以分步解决，整体复杂度与每一步的组合数是乘积的关系。通过分步、缩小每一步的组合数，可以有效降低问题复杂度。

2.2.1 单车运输的全部可行方案

所谓单车运输的全部可行方案，指：我们认为，存在一个全集，不管选用多少量车运输，每一辆车的运输方案都属于该集合。

我们可以通过尽可能的缩小该集合，然后排列组合得出不重不漏的所有运输方案的集合。最佳的运输方案必然在这个不重不漏的集合中。

简单证明如下：

- 漏掉用户，方案不满足基本要求。
- 客户重复，因为两点间直线距离最短，案例中两两节点之间均直接相连。客户重复必然不是最短路程和最短时间。

对于 8 客户节点规模的问题，单车运输全部可行方案的最复杂情形是: $2^8 = 256$

若加上载重量、客户要求时间等因素的约束，该运输方案全集的平均复杂度小于 100.

2.3 多车全量覆盖客户的可行运输方案

该部的时间复杂度与单车运输方案的可行解个数直接相关。

针对本题的数据，通过简单的估算，单车运输方案的可行解个数可以优化到50 个以下。

最差情形的时间复杂度是: 2^50

通过剪枝优化等算法，我们可以尝试减掉大部分分支，将问题规模减少到一个可以接受的范围。

2.3.1 全部可行解集合中选最优解

该部不存在较大的时间复杂度问题，与可行解的集合数是线性关系。无需特殊优化。

3 模型假设

1. 只要物流中心的车辆足够多（极限值是为每一个客户提供一辆专用的运输车辆），存在所有客户均不迟到的解决方案。
2. 客户不反对提前到达，但，即使提前到达，也要求在客户规定的时间范围内开始卸货。

3. 路程对成本和最终决策的影响，远远大于等待时间。

4 符号设定

符号	说明
S_i	货车i 经过的客户序列
$D_{i,i+1}$	货车i 经过的客户序列中，第 i 和 $i+1$ 个客户之间的距离
X_{S_i}	序列 S_i 的总路程
T_{latei}	货车i 迟到的总时间
T_{early_i}	货车i 提前到达等待的总时间
a_i	客户 i 要求到达的最早时间
b_i	客户 i 要求到达的最晚时间
q_i	客户 i 所需物资吨数
onlyPositive(x)	若 $x < 0$ 则等于0，否则等于 x。

5 模型建立

根据问题分析的思路，该问题可以按照一个多目标规划问题来处理。
我们根据因素和目标分别写出目标函数和约束条件，最后做统一汇总。

5.1 路程

假设 i 车运输的客户序列为: S_i，X_{S_i} 是序列 S_i 经过的总路程，序列 S_i 有 n_i 个客户。

路程越小越好，所以，目标函数取最小值。

所以有:

$$target \quad : \quad \min = \sum_i X_{S_i} \tag{1}$$

$$constraint \quad : \quad X_{S_i} = \sum_{j=1}^{n_i} D_{j-1,j} + D_{n_i,0} \tag{2}$$

因为火车最终需要回到物流中心，所以，式(2) 需要加最后一项。

5.2 迟到时间

假设 i 车运输的客户序列为: S_i，T_{latei} 是序列 S_i 迟到的总时间，序列 S_i 有 n_i 个客户。假设货车 i 从物流中心出发的时间为 x_i

所以有:

$$target \quad : \quad \min = \sum_i T_{latei} \tag{3}$$

$$constraint \quad : \quad T_{latex1} = OnlyPositivex + \frac{D_{0,1}}{50} - b_1 \tag{4}$$

$$T_{latei} = OnlyPositiveb_{i-1} + T_{latexi-1} + + \frac{D_{0,1}}{50} - b_1 \tag{5}$$

onlyPositive(x), 若 $x < 0$ 则等于0，否则等于 x。
其中式(5) 是计算迟到时间的递推规则，式(4) 是边界条件

5.3 提前到达等待

假设 i 车运输的客户序列为: S_i，T_{early_i} 是序列 S_i 迟到的总时间，序列 S_i 有 n_i 个客户。假设货车 i 从物流中心出发的时间为 x_i

所以有:

$$\begin{aligned} target &: \min = \sum_i T_{early_i} && (6) \\ constraint &: T_{early_1} = OnlyPositive x + \frac{D_{0,1}}{50} - a_1 && (7) \\ & T_{latei} = OnlyPositive a_{i-1} + T_{early_{i-1}} + + \frac{D_{0,1}}{50} - a_1 && (8) \end{aligned}$$

onlyPositive(x), 若 $x < 0$ 则等于0，否则等于 x。

其中式(8) 是计算等待时间的递推规则，式(7) 是边界条件。

5.4 全部时间消耗

已经分别考虑了迟到时间、等待时间、运输时间（与运输距离线性相关），此处不做重复计算，以免影响结果的因子权重。

5.5 货车载重量

仅需作为约束条件：

$$\sum_j \delta_{ij} q_j \leq 8 \qquad (9)$$

其中，$\delta_{ij} = 0$ 表示客户 j 由货车 i 运输。

5.6 货车数量

作为目标函数：

$$target : \min = n \qquad (10)$$

5.7 目标函数与约束汇总

$$\begin{aligned} target &: \min = \sum_i X_{S_i} && (11) \\ & \min = n && (12) \\ & \min = \sum_i T_{early_i} && (13) \\ & \min = \sum_i T_{latei} && (14) \\ constraint &: T_{latex1} = OnlyPositive x + \frac{D_{0,1}}{50} - b_1 && (15) \\ & T_{latei} = OnlyPositive b_{i-1} + T_{latexi-1} + + \frac{D_{0,1}}{50} - b_1 && (16) \end{aligned}$$

$$\sum_j \delta_{ij} q_j \leq 8 \tag{17}$$

$$X_{S_i} = \sum_{j=1}^{n_i} D_{j-1,j} + D_{n_i,0} \tag{18}$$

$$T_{early_1} = OnlyPositivex + \frac{D_{0,1}}{50} - a_1 \tag{19}$$

$$T_{late i} = OnlyPositivea_{i-1} + T_{early_{i-1}} + + \frac{D_{0,1}}{50} - a_1 \tag{20}$$

6 模型求解

该模型的求解，可以使用lingo 直接求解，但lingo 求解的本质也是遍历可行解选最优。lingo 做遍历时，无法针对具体问题对可行解进行优化。

我们使用 python 编程解决该问题。 matlab 的优势是数值计算，本题的求解过程主要涉及逻辑处理，数值计算的特征不明显。python 更适合于处理该问题。

6.1 单车运输的全部可行方案–优化至32 个可行方案

6.1.1 最坏情形

最坏情形即二项分布的情形，8 个客户共 2^8 种组合。每一种组合内部还要进行一次全排列作为运输序列。

6.1.2 载重量优化

使用约束条件式(17) 缩小可行解的范围，得到 65 种组合。每一种组合内部依旧要进行一次全排列作为运输序列。

6.1.3 时间序列优化

我们采用严格遵守客户时间要求的方式筛选方案，
即将目标函数化为约束，如下：

$$0 = \sum_i T_{early_i} \tag{21}$$

$$0 = \sum_i T_{late i} \tag{22}$$

此处对 65 个组合分别进行全排列，取全部可行解。最终共计 32 个序列。

6.2 多车不重不漏覆盖全部客户的可行解–333组可行解

利用上一步得到的 32 个可行序列，使用组合排列的方式寻找全部可行解，约束条件为覆盖客户不重不漏。得到 333 个可行解。

6.3 以路程和等待时间作为目标函数选取最优解

在之前的处理中，采用严格遵循客户要求的原则，等待时间已经作为约束处理，此处不做重复加权。

仅以最短路程作为目标函数求解最优解。

得到的解为：

车辆编号	负责的用户序列	最佳出发时间	总路程

7 模型评价和进一步扩展

们给出的物流规划模型是多目标规划模型，主要将路程、客户对时间的要求作为规划的目标，将载重量等作为约束进行处理。

模型求解时，遵循严格遵循客户需求的原则，所以，客户对时间要求作为约束需要严格遵循。从而实现了化多目标为单目标。后续可以使用相对简单一些的单目标规划问题求解思路进行求解。

模型求解时，针对该类规划问题的特殊性，我们可以分步进行。首先计算单车运输的全部可行方案，经过剪枝处理后，只有 32 组可行方案。最后给出的多车全量覆盖客户的可行运输方案由这 32 组方案中的任意 n 组组合而成。问题规模大幅度缩小，从而实现了在短时间内计算出全局最优解。