

ljskljklkkjkjkj s

摘要

hello

1 问题重述

1.1 提出问题

近年来我国汽车市场每年平均以 25% 的速度快速增长，其发展速度快于国民经济的增长。汽车消费需求的增长，促进了乘用车的整车物流量迅速上升。乘用车生产厂家根据全国客户的购车订单，向物流公司下达运输乘用车到全国各地的任务，物流公司则根据下达的任务制定运输计划并配送这批乘用车。为此，物流公司首先要从他们当时可以调用的“轿运车”中选择出若干辆轿运车，进而给出其中每一辆轿运车上乘用车的装载方案和目的地，以保证运输任务的完成。但因轿运车、乘用车规格的多样性和运输路线的多样性，当前很多物流公司在制定运输计划时主要依赖调度人员的经验。而在运输任务较复杂时，往往会发生运输效率低下，运输成本不理想等问题。物流公司希望能够在确保完成运输任务的前提下寻求降低运输成本的方法。“轿运车”是通过公路来运输乘用车整车的专用运输车。本文涉及的双层轿运车分为三种子型：上下层各装载 1 列乘用车，故记为 1-1 型（图 1）；下、上层分别装载 1、2 列，记为 1-2 型（图 2）；上、下层各装载 2 列，记为 2-2 型（图 3），每辆轿运车可以装载乘用车的最大数量在 6 到 27 辆之间。



图 1: 1-1 型轿运车



图 2: 1-2 型轿运车



图 3: 2-2 型轿运车

1.2 问题要求

建立数学模型，以通用算法和程序解决整车物流成本优化的问题，能够比较准确地模拟整车物流的装载方案及行车路线，更好地降低整车物流运输成本。问题一：物流公司要运输 I 车型的乘用车 100 辆及 II 车型的乘用车 68 辆。问题二：物流公司要运输 II 车型的乘用车 72 辆及 III 车型的乘用车 52 辆。问题三：物流公司要运输 I 车型的乘用车 156 辆、II 车型的乘用车 102 辆及 III 车型的乘用车 39 辆。问题四：物流公司要运输 166 辆 I 车型的乘用车（其中目的地是 A、B、C、D 的分别为 42、50、33、41 辆）和 78 辆 II 车型的乘用车（其中目的地是 A、C 的，分别为 31、47 辆），具体路线见图 4，各段长度：OD=160，DC=76，DA=200，DB=120，BE=104，AE=60。问题五：附件的表 1 给出了物流公司需要运输的乘用车类型（含序号）、尺寸大小、数量和目的地，附件的表 2 给出可以调用的轿

运车类型（含序号）、数量和装载区域大小（表里数据是下层装载区域的长和宽, 1-1 型及 2-2 型轿运车上、下层装载区域相同；1-2 型轿运车上、下层装载区域长度相同，但上层比下层宽 0.8 米。此外 2-2 型轿运车因为层高较低，上、下层均不能装载高度超过 1.7 米的乘用车。

2 模型假设

1. 题目中所列数据均真实可靠；
2. 乘用车与轿运车两侧的安全距离在说明装载区域大小时已经考虑；
3. 乘用车不可在轿运车上倾斜摆放；
4. 轿运车运输路线为有向图；
5. 运输花费主要由轿运车数量决定，次要由轿运车类型决定。

3 基本符号说明

N	乘用车总数
K	为轿运车使用量
w	乘用车宽度

- N 乘用车总数
- i 乘用车下标, $i \in N$
- K 为轿运车使用量
- k 轿运车下标, $k \in K$
- j 轿运车层级下标, $j \mapsto \{1, 2\}$
- l 乘用车长度
- l_i 第 i 辆乘用车长度
- w 乘用车宽度
- w_i 第 i 辆乘用车的宽度
- h 乘用车高度
- h_i 第 i 辆乘用车的高度
- C 运输成本
- C_k 第 k 辆轿运车的运输成本

- L 轿运车长度
- L_{jk} 第 k 辆轿运车第 j 层的长度
- W 轿运车宽度
- W_{jk} 第 k 辆轿运车第 j 层的宽度
- T 轿运车类型
- T_k 第 k 辆轿运车的类型, $T_k \mapsto \{1-1, 1-2, 2-2 \dots\}$

4 问题分析

本文的基本问题是解决如何分配乘用车到轿运车并使整体运输成本最低的问题。经过对问题本身的分析, 该问题可以看左组合优化问题 (Combinatorial Optimization Problem), 而组合优化问题从计算复杂度上来看属于 NP-困难问题, 即在多项式时间内无法确定能够找到有效算法求解的问题。

本文拟采用约束编程 (Constraint Programming) 的思想进行建模与求解。

而对于组合优化的问题, 还需列出对应的最大化或者最小化的目标函数。由于本文所提乘用车物流运输计划的问题从计算复杂度上来看属于 NP-困难, 所以在解决的过程中, 引入启发式算法 (Heuristics)。

4.1 技术背景

4.1.1 约束编程

约束编程是一种编程范型, 在这种编程范型中, 变量之间的关系以约束的形式组织。一般地, 约束编程的支持通用的求解程序对约束模型进行求解

Gecode Gecode[1] 是一个基于 c++ 实现的约束编程模型求解器。

cumulative 约束 cumulative 约束是一类可以解决在一定资源情限制清况下任务分配的约束。任务用 4-元组 (起始时间, 持续时间, 终止时间, 占用资源) 表示, 而资源用 2-元组 (资源, 限制) 表示。对于给定的任务集合 \mathcal{T} , 在单资源的情况下, cumulative 约束强制在任一时间点, 任务所占的资源数不超过所限制的阈值, cumulative 约束所适用的问题如图 4所示。

而对于多资源的情况, 即对一组资源 \mathcal{R} 和一组任务 \mathcal{T} , 任务必须被分配到可用的资源并且对各资源的使用不超过资源的限制。定义多资源情况下的 cumulative 约束原型如下:

$$cumulatives(resource, start, duration, end, height, limit)$$

对于本文所提乘用车物流运输计划问题, 关键是解决乘用车的分配问题。抽象地看, 可以有效利用约束编程中 cumulative 约束

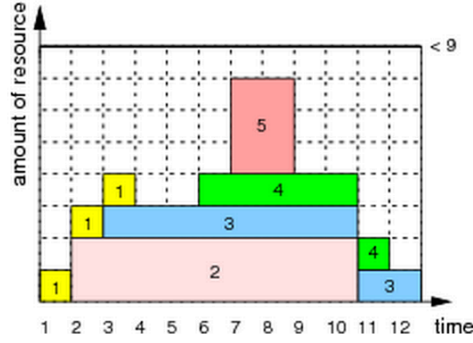


图 4: cumulative 约束

4.2 问题一、二、三分析：

问题一、二、三要求计算各种类型轿运车的数量和每辆轿运车的乘用车装载方案，各类型乘用车数量已知，所需各类型轿运车数量及各轿运车的装载方案未知，但不涉及运输目的地的差异。并且，问题一、二、三的差异仅为输入参数不同，所以拟建立通用的模型进行求解。

4.2.1 模型建立

基本思想

模型参数 本模型涉及模型基本参数如前所述

决策变量

- $s_k \mapsto \{0,1\}$, $k \in K$, 若变量 s_k 为 1, 则表示第 k 辆轿运车已经被使用; 反之, 则第 k 辆轿运车未被使用。
- $x_{ijk} \mapsto \{0,1\}$, $i \in N$, $k \in K$, $j \in \{1,2\}$, 若 x_{ijk} 为 1, 则表示第 i 辆乘用车被分配到了第 k 辆轿运车的第 j 层。

目标函数 本问题的目标是最小化已使用的轿运车的数量, 而考虑到题设中 当所使用的轿运车数量相同的, 优先使用耗油量较少的车, 所以在考虑目标函数时, 我们采用对每种不同的车型赋不同的成本值, 即在考虑 1-1 型, 1-2 型和 2-2 型轿运车的情况下, $C_{1-1}=1$, $C_{1-2}=1.0002$, $C_{2-2}=1.0004$ 。通过微调不同车型的对应成本值, 可以保证在优化过程中能够首先考虑轿运车的使用数目, 同时在使用数目相同的情况下使用耗油量更少的轿运车。

$$C = \sum_{k=1}^K C_k \cdot s_k \quad (1)$$

约束条件

- 只有当轿运车确定为使用状态时才能为其分配乘用车

$$x_{ijk} \leq s_k \quad \forall i, j \quad (2)$$

- 所有乘用车必须被分配给某一轿运车

$$\sum_{k=1}^K \sum_{j=1}^2 x_{ijk} = 1 \quad \forall i \quad (3)$$

- 1-2 的使用数目不得超过 1-1 使用数目的 20%

$$\sum_{k=1}^K s_k[T_k = 2] \leq 20\% \cdot \sum_{k=1}^K s_k[T_k = 1] \quad (4)$$

- 对于高度超过 1.7 米的乘用车，只能放在 1-1 或 1-2 型轿运车的下层

$$\forall i \quad h_i > 1.7 \implies \sum_{k=1}^K x_{i1k} = 0 \quad (5)$$

- 对于任一轿运车的任一层，所包含的乘用车必须能够满足该区域的长宽要求

$$\forall k, j \quad x_{ijk} = 1 \implies \text{cumulatives}(W_{jk}, \{x_{coord_i} | x_{ijk} = 1\}, \{w_i | x_{ijk} = 1\}, \{x_{coord_i} | x_{ijk} = 1\}, \{l_i | x_{ijk} = 1\}, L_{jk}) \quad (6)$$

4.3 问题四分析

5 模型的建立与求解

5.1 问题一、二、三通用

5.1.1 模型建立

实例变量

决策变量

约束条件

5.1.2 模型求解

本模型是基于约束编程的思想所建立, 模型的求解使用 Gecode [1]

6 模型的评价、改进及推广

参考文献

[1] Gecode Team. *Gecode: Generic Constraint Development Environment*, 2006.

A 代码