参赛密码 _____

（由组委会填写）



第十一届华为杯全国研究生数学建模竞赛

题目 乘用车物流运输计划问题

摘　　要：

本文对乘用车物流运输计划问题进行了研究，针对不同情境，将问题抽象成不同维度的箱柜装载问题进行建模，并取得了以下几方面的成果。

针对问题 1,2,3： 采用约束编程的建模思想，建立了适用于多种尺寸的乘用车、轿运车的 3 维乘用车装载数学模型，并获得了在给定乘用车车型数量的前提下，最优的运载分配策略。

结论（1）：在题中给定的约束条件下，运载 100 辆 I 车型的乘用车及 68 辆 II 车型的乘用车，至少需 16 辆 1-1 型乘用车及 2 辆 1-2 型乘用车。

结论（2）：在题中给定的约束条件下，运载 72 辆 II 车型的乘用车及 52 辆 III 车型的乘用车，至少需。。。辆 1-1 型乘用车及。。。辆 1-2 型乘用车。

结论（3）：在题中给定的约束条件下，运载 156 辆 I 车型的乘用车及 102 辆 II 车型的乘用车，至少需。。。辆 1-1 型乘用车及。。。辆 1-2 型乘用车。

针对问题 4： 进一步优化了问题 1,2,3 中的数学模型，完善并支持了考虑行车路线问题的轿运车运输乘用车问题，并给出了通用算法和程序解决了乘用车装载方案及行车路线方案。结论：在题中给定的约束

条件及运输路线下，运输 166 辆 I 车型的乘用车和 78 辆 II 车型的乘用车至少需。。。辆 1-1 型乘用车及。。。辆 1-2 型乘用车。

针对问题 5： 在乘用车、轿运车类型、尺寸、数量、目的地更多样化的情境下，持续优化了问题 4 中建立的 3 维乘用车装载模型，以支持复杂的乘用车装载问题的建模和分析需求，并通过经验分析增加了必要的约束条件以缩小解空间，在更短时间内求得较优解。结论：在题中给定的约束条件下，运载附件中所示乘用车到指定目的地，至少需。。。辆 1-1 型乘用车及。。。辆 1-2 型乘用车。本文综合运用数理统计、分支限界等数学方法，箱柜装载问题的建模思想，借助 Gecode 系统建立了多种满足约束编程思想的数学模型，并对所提出问题进行了研究，具有很好的实用性与推广性。最后，总结了模型的优点与不足，为后续研究此类问题的学者提供了一个新的思路。

目录

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，能够比较准确得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
- 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
- d 乘用车目的地
- d_i 第 i 辆的目的地 $d_i \mapsto \{A, B, C, D\}$
- D 表示出发地 O 到各目的地的距离，$D = \{D_A, D_B, D_C, D_D\}$
- 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，$T_k \mapsto \{1,\ 2,\ 3\ldots\}$，其中 1 代表 1-1 型轿运车，2 代表 1-2 型轿运车，3 代表 2-2 型轿运车

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，该问题可以视作组合优化问题 (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。

4.1 问题一、二、三分析：

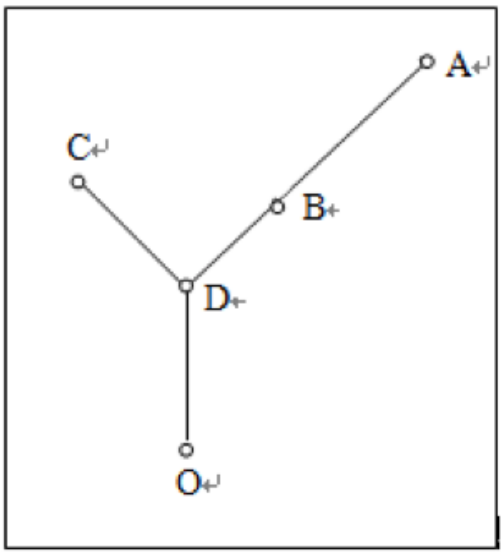
问题一、二、三要求计算各种类型轿运车的数量和每辆轿运车的乘用车装载方案，各类型乘用车数量已知，所需各类型轿运车数量及各轿运车的装载方案未知，但不涉及运输目的地的差异。并且，问题一、二、三的差异仅为输入参数不同，所以拟建立通用的基于约束编程的模型进行求解。解决该问题一共有三个关键点：

- 乘用车如何分配给轿运车；分配给轿运车的何层。
- 对于已分配的乘用车，是否满足该轿运车层的长宽约束。
- 采用何种搜索策略能够

4.2 问题四分析

问题四要求在考虑行车路线的情况下解决轿运车运输乘用车问题，计算所需各类型轿运车的数量及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。为解决这一问题，需要在前三题建立的通用模型基础之上，额外考虑运输路线的因素，并添加相关的限制条件。因为我们假设运输路线是一个有向图，所以只会有 O->D->C, 和 O->D->B->A 这两条运输路线。因此，针对路线 O->D->C，可以约束只有终点是 D，C 的乘用车可以在一辆轿运车上一次运输。针对路线 O->D->B->A，可以约束只有终点是 D，B，A 的乘用车可以在一辆轿运车上一次运输。因终点是 D 的乘用车在这两条路线的轿运车里都可以装载，因此可以把装载终点是 D 的乘用车的装载优先级放低。又因行驶路线为有向图，所以在 O->D->C 这一运输路线中，我们认为终点是 C 的乘用车的运载优先级高于终点是 D 的乘用车。同理可得，终点是 A 的乘用车的运载优先级高于终点是 B 的乘用车，而终点是 B 的乘用车的运载优先级高于终点是 D 的乘用车。各段长度：OD=160，

DC=76，DA=200，DB=120 也需要被添加到约束条件里，以将行使里程考虑在行使成本中，半定量化最终的运输成本。因此，对这一问，可以在解决 1-3 问的通用数学模型的基础之上，加上以上这些约束条件以保证满足本题的运载条件，并获得在给定条件下的最优解。



5 技术背景

5.1 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，ke 可用于开发基于约束的系统应用程序，是一个可移植、高效的环境。Gecode 是从根本上进行编程开放，这意味着它可以很容易地与其他系统的接口。它支持新的传播者（如约束的执行情况），分支策略，和搜索引擎编程。新的变量域可以被编程的效率，在有限域和整数集，拿出 Gecode 预定义变量相同的水平。

6 模型的建立与求解

6.1 模型建立

6.1.1 问题一、二、三通用模型

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。

$$C = \sum_{k=1}^{K} C_k \cdot S_k \tag{1}$$

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

$$x_{ijk} \le s_k \quad \forall i,\ j \tag{2}$$

- 所有乘用车必须被分配给某一轿运车

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

- 1-2 的使用数目不得超过 1-1 使用数目的 20%

$$\sum_{k=1}^{K} s_k[T_k = 2] \le 20\% \cdot \sum_{k=1}^{K} s_k[T_k = 1] \tag{4}$$

- 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

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

- 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

$$\begin{aligned} &\forall k, j \quad x_{ijk} = 1 \implies \\ &cumulatives(W_{jk}, \{xcoord_i | x_{ijk} = 1\}, \{w_i | x_{ijk} = 1\}, \{xcoord_i | x_{ijk} = 1\}, \{l_i | x_{ijk} = 1\}, L_{jk}) \end{aligned} \tag{6}$$

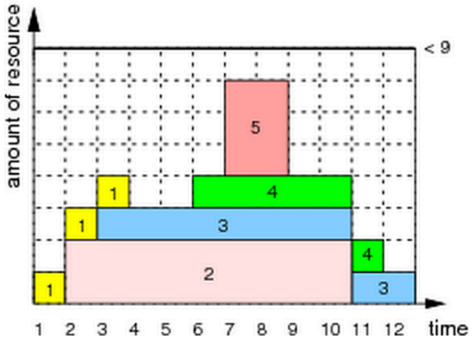


图 4: cumulative 约束

cumulative 约束 在约束编程的范型中，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 $\mathscr{T}$，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，单资源情况下 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 $\mathscr{R}$ 和一组任务 $\mathscr{T}$，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：

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

根据 [2] 中所述思想，我们采用了基于交换的 cumulatives 约束实现，具体算法如图 5和图 6所示：

```
1   Set nb_task_r, sum_height_r and top_prune_r to 0.
2   Extract the next event &type,task,date,increment&
3   Set d to date.
4   while type,task,date,increment≠NULL do
5     if type≠pruning then
6       if d≠date then
7         if nb_task_r>0 and sum_height_r<Limit[r] then fail.
8         Set d to date.
9       if type=check then Add increment to nb_task_r else Add increment to
          sum_height_r
10    else Set top_prune_r to top_prune_r+1 and set stack_prune_r[top_prune_r] to task.
11    Extract the next event &type,task,date,increment&
12  if nb_task_r>0 and sum_height_r<Limit[r] then fail.
```

图 5: 交换算法 1

```
1   if nb_task_r≠0 and sum_height_r−contribution[t] < Limit[r] then.
2       if  FIXVAR(Machine[t], r)=fail
3       or ADJUSTMINVAR(Origin[t], up−max(Duration[t])+1)=fail
4       or ADJUSTMAXVAR(Origin[t], low)=fail
5       or ADJUSTMAXVAR(End[t], low+ max(Duration[t]))=fail
6       or ADJUSTMINVAR(End[t], up+1)=fail
7       or ADJUSTMINVAR(Duration[t], min(up−max(Origin[t])+1,
min(End[t])−low))=fail then fail.
```

图 6: 交换算法 2

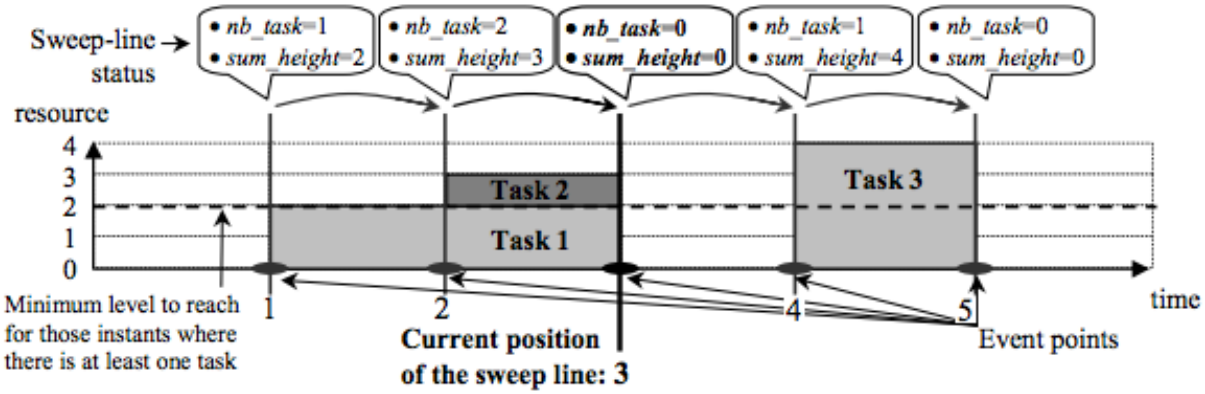


图 7: 交换算法示例

搜索策略 为了在解空间的搜索过程中不断最小化目标函数寻找最优解，本问题采用分支定界法（Branch and Bound）[3] 来寻找模型的最优解。

它是由 R.J. 达金和兰德－多伊格在 20 世纪 60 年代初提出的。分支定界法的基本思想是对有约束条件的最优化问题的所有可行解（数目有限）空间进行搜索。所谓“分支”是采用广度优先的策略，依次生成扩展结点的所有分支（即：儿子结点）。所谓“限界”是在结点扩展过程中，计算结点的上界（或下界），边搜索边减掉搜索树的某些分支，从而提高搜索效率。该算法在具体执行时，把全部可行的解空间不断分割为越来越小的子集（称为分支），并为每个子集内的解的值计算一个下界或上界（称为定界）。在每次分支后，对凡是界限超出已知可行解值的那些子集不再做进一步分支。这样，解的许多子集（即搜索树上的许多结点）就可以不予考虑了，从而缩小了搜索范围。这一过程一直进行到找出可行解为止，该可行解的值不大于任何子集的界限，因此这种算法一般可以求得最优解。

分枝界限法是组合优化问题的有效求解方法，其步骤如下所述：

- 步骤一：如果问题的目标为最小化，则设定目前最优解的值 Z=∞
- 步骤二：根据分枝法则（Branching rule），从尚未被洞悉（Fathomed）节点（局部解）中选择一

个节点，并在此节点的下一阶层中分为几个新的节点。

- 步骤三：计算每一个新分枝出来的节点的下限值（Lower bound，LB）
- 步骤四：对每一节点进行洞悉条件测试，若节点满足以下任意一个条件，则此节点可洞悉而不再被考虑：此节点的下限值大于等于 Z 值。已找到在此节点中，具最小下限值的可行解；若此条件成立，则需比较此可行解与 Z 值，若前者较小，则需更新 Z 值，以此为可行解的值。此节点不可能包含可行解。
- 步骤五：判断是否仍有尚未被洞悉的节点，如果有，则进行步骤二，如果已无尚未被洞悉的节点，则演算停止，并得到最优解。

6.1.2 问题四模型

6.1.3 问题五模型

6.2 模型求解

本文所有涉及模型均采用 Gecode[1] 求解

6.2.1 问题一、二、三求解

轿用车类型（第五问是序号）	相同类型、相同装载方式的车辆数	装在上层序号为 1 乘用车数量	装在上层序号为 2 乘用车数量	装在上层序号为 3 乘用车数量	装在下层序号为 1 乘用车数量	装在下层序号为 2 乘用车数量	装在下层序号为 3 乘用车数量	中间停靠地	目的地
1-1	5	4	0	0	4	0	0	0	0
1-2	1	10	0	0	5	0	0	0	0
1-1	5	4	0	0	4	0	0	0	0
1-2	1	0	12	0	5	0	0	0	0
1-1	5	0	5	0	0	5	0	0	0
1-1	1	0	1	0	0	5	0	0	0

问题一求解

问题二求解

问题三求解

6.2.2 问题四求解

6.2.3 问题五求解

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] 16,2，
25,5 22,4

轿用车类型	相同类型、相同装载方式的车辆数	装在上层序号为 1 乘用车数量	装在上层序号为 2 乘用车数量	装在下层序号为 1 乘用车数量	装在下层序号为 2 乘用车数量	中间停靠地	目的地
1-1	5	4	0	4	0	0	A
1-1	1	0	5	0	5	0	A
1-2	1	2	10	0	6	0	A
1-1	1	4	0	0	5	B	A
1-1	3	4	0	4	0	0	B
1-2	1	10	0	5	0	0	B
1-1	1	4	0	4	0	D	B
1-1	4	4	0	4	0	0	C
1-2	1	1	11	0	6	0	C
1-1	3	0	5	0	5	0	C
1-1	3	4	0	4	0	0	D
1-2	1	10	0	5	0	0	D
1-1	1	0	0	1	0	0	D

7 模型的评价、改进及推广

参考文献

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

[2] Beldiceanu, Nicolas, and Mats Carlsson. "A new multi-resource cumulatives constraint with negative heights." Principles and Practice of Constraint Programming-CP 2002. Springer Berlin Heidelberg, 2002.

[3] Land, Ailsa H., and Alison G. Doig. "An automatic method of solving discrete programming problems." Econometrica: Journal of the Econometric Society (1960): 497-520.

A 代码