

# 基于 cvxpy 线性最优化的平板车装货问题模型

## 摘要

针对两辆平板车装货问题，本文使用了 Python 最优化工具库 cvxpy，基于多约束条件下的线性规划问题进行了建模与求解，解得每辆平板车分别装载的 7 类包装箱个数列为 4、1、5、3、3、2、0 与 4、6、4、3、0、1、0，总浪费空间为 0.006m（按厚度计），空间利用率达 99.97%。

## 一、问题重述

有七种规格的包装箱要装到两辆铁路平板车上去。包装箱的宽和高是一样的，但厚度( $t$ ，以厘米计)及重量( $w$ ，以公斤计)是不同的。下表给出了每种包装箱的厚度、重量以及数量。每辆平板车有 10.2 米长的地方可用来装包装箱（像面包片那样），载重为 40 吨。由于当地货运的限制，对  $C_5, C_6, C_7$  类的包装箱的总数有一个特别的限制：这类箱子所占的空间（厚度）不能超过 302.7cm。

问题：制定数学模型，求解如何装配包装箱使得包装箱装到平板车上去使得浪费的空间最小。

表 1. 包装箱规格与数量

|         | $C_1$ | $C_2$ | $C_3$ | $C_4$ | $C_5$ | $C_6$ | $C_7$ |
|---------|-------|-------|-------|-------|-------|-------|-------|
| $t(cm)$ | 48.7  | 52.0  | 61.3  | 72.0  | 48.7  | 52.0  | 64.0  |
| $w(kg)$ | 2000  | 3000  | 1000  | 500   | 4000  | 2000  | 1000  |
| 件数      | 8     | 7     | 9     | 6     | 6     | 4     | 8     |

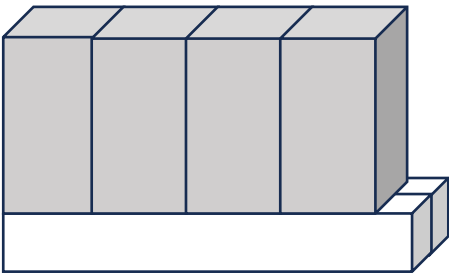


图 1. 包装箱装配示意图

## 二、模型假设

- 1.不考虑包装箱在平板车上的装配顺序；
- 2.包装箱装配在平板车上时，箱与箱间不存在空隙；
- 3.包装箱的高度、宽度对本问题无影响。

## 三、符号定义

表 2：文中用到的符号定义

|       |   |
|-------|---|
| $t_i$ | 第 <i>i</i> 类包装箱厚度（cm）( $0 < i \leq 7$ ) |
|-------|---|

|                |   |
|----------------|---|
| $C_i$          | 第 <i>i</i> 类包装箱 (0 < <i>i</i> ≤ 7)          |
| $d_i$          | 第 <i>i</i> 类包装箱总件数 (0 < <i>i</i> ≤ 7)       |
| $w_i$          | 第 <i>i</i> 类包装箱重量 (kg) (0 < <i>i</i> ≤ 7)   |
| $x_i$          | 第一辆平板车上第 <i>i</i> 类包装箱数量 (0 < <i>i</i> ≤ 7) |
| $y_i$          | 第二辆平板车上第 <i>i</i> 类包装箱数量 (0 < <i>i</i> ≤ 7) |
| $Weight$       | 每辆平板车载重限制 (40t)                             |
| $Length$       | 每辆平板车长度限制 (10.2m)                           |
| $Length_{567}$ | 5、6、7 类包装箱总厚度限制 (302.7m)                    |
| $N$            | 自然数集  |
| $Target$       | 目标函数, 满足约束条件的最大装配厚度                         |

## 四、问题分析

本问题为多重约束条件下的线性规划问题, 需要在满足总重量、总厚度、部分厚度等限制的情况下, 给出装配包装箱到两辆平板车上的具体方案, 使两辆车总剩余空间最小。因此可采取线性最优化方案及相关工具求解。

## 五、模型的建立与求解

根据题意, 得出如下数学模型

目标函数:

$$Target = \max \left( \sum_{i=1}^7 t_i x_i + \sum_{i=1}^7 t_i y_i \right) \quad (1)$$

约束条件:

$$\begin{cases} \sum_{i=1}^7 t_i x_i \leq Length \\ \sum_{i=1}^7 t_i y_i \leq Length \\ \sum_{i=1}^7 w_i x_i \leq Weight \\ \sum_{i=1}^7 w_i y_i \leq Weight \\ \sum_{i=5}^7 t_i x_i + \sum_{i=5}^7 t_i y_i \leq Length_{567} \\ x_i + y_i \leq d_i, i = 1, 2, 3, 4, 5, 6, 7 \\ x_i, y_i \in N, i = 1, 2, 3, 4, 5, 6, 7 \end{cases} \quad (2)$$

使用基于 Python 语言的最优化工具库 cvxpy 求解，结果如下：

表 3: cvxpy 求解结果

|       |   |   |   |   |   |   |   |
|-------|---|---|---|---|---|---|---|
| $i$   | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| $x_i$ | 4 | 1 | 5 | 3 | 3 | 2 | 0 |
| $y_i$ | 4 | 6 | 4 | 3 | 0 | 1 | 0 |

$Target = 20.394m$ ，即浪费空间为  $0.006m$ ，为最小浪费厚度。

经检验，该结果满足(2)中所有约束条件，利用率达到 99.97%。

## 六、模型总结与推广

1. 本模型使用了基于多重约束的线性规划问题的建模解决方案，建立了整数规划模型；
2. 使用基于 Python 语言的最优化工具库 cvxpy 进行求解，高效、准确；
3. 本问题建立的数学模型求解实现方式多样，除 cvxpy 外，也可基于 Lingo、Matlab 等多种工具进行求解；
4. 模型可类比推广以解决其他的简单最优化问题，如求解装配重量最大化问题等；
5. 模型不适用于涉及到装配成本最小化等复杂多因素最优化问题，需要进行进一步改进。

## 参考文献

- [1] 【Python】线性规划问题求解(cvxpy),[https://blog.csdn.net/liqian\\_ken/article/details/119605243](https://blog.csdn.net/liqian_ken/article/details/119605243)
- [2] CVXPY 1.3 documentation,<https://www.cvxpy.org/#>
- [3] 第 4 讲 线性规划，《赵静数学建模第四版》

## 附录

Python 源代码：见“附件 1”。