

# 基于遗传算法的多维装箱问题的研究

绍兴文理学院元培学院 张森均

【摘要】装箱问题属于典型的NP完全问题，旨在充分考虑容器承重、容积等因素的前提下，对装箱问题通过数学建模与算法实现来提高空间利用率，从而实现利润最大化。本文主要对多维空间的装箱过程展开研究，并基于遗传算法进行建模和分析。

【关键词】装箱问题；调度；启发式算法；遗传算法

DOI:10.19353/j.cnki.dzsj.2017.10.053

## 1 引言

在满足车辆限载条件下，将同一时间、同一路线上的货物进行集中安排配送车辆，使每辆车的容积利用率达到最大。为了便于装车，简化装箱复杂度，本文将装箱问题设计为二次装箱模式。第一次装箱：在按需包装环节，将目的地区域相同，尺寸小于包装箱、允许混合装箱的货物进行合并装箱。统一装箱后，货物之间的相互挤压将转变为外包装箱体之间的挤压，同时减少货物的装箱件数，方便货物的派送和装卸。

## 2 装箱问题的数学模型

根据模型求解需要，坐标轴x轴y轴z轴为容器的长宽高，各符号说明如表1：

表1 符号说明

符号	符号说明	符号	符号说明
$CU$	代表货物 1	$V$	容器的体积
$L_i$	货物 1 的长度	$E$	容器的剩余空间
$W_i$	货物 1 的宽度	$\mathcal{R}_i$	$(0, 1)$ 变量， $CU$ 装入容器中时值为 1
$H_i$	货物 1 的高度	$\mathcal{R}_i^{yz}$	$(0, 1)$ 变量， $CU$ 装入容器中时值为 1，否则为 0
$L$	容器的长度	$PL_i^x$	$(0, 1)$ 变量， $CU$ 沿 x 轴方向中点 $W_i$ 时值为 1，否则为 0
$W$	容器的宽度	$PW_i^x$	$(0, 1)$ 变量， $CU$ 沿 y 轴方向中点 $W_i$ 时值为 1，否则为 0
$H$	容器的高度	$PL_i^z$	$(0, 1)$ 变量， $CU$ 沿 z 轴方向中点 $H_i$ 时值为 1，否则为 0

数学规划模型如下：

$$\min E = V - \sum_{i=1}^n (\mathcal{R}_i * V_i) \quad (1)$$

$$s.t. \sum_{i=1}^n \mathcal{R}_i^{yz} \leq 1, \forall x \in \{1, 2, \dots, L\}, y \in \{1, 2, \dots, W\}, z \in \{1, 2, \dots, H\} \quad (2)$$

$$\sum_{i=1}^n \mathcal{R}_i^{yz} - \mathcal{R}_i^{yz(i+1)} \geq 0, \forall x \in \{1, 2, \dots, L\}, y \in \{1, 2, \dots, W\}, z \in \{1, 2, \dots, H-1\} \quad (3)$$

$$x_i + \mathcal{R}_i * (PL_i^x * L_i + PW_i^x * W_i) \leq L, \forall i \in \{1, 2, \dots, n\} \quad (4)$$

$$y_i + \mathcal{R}_i * (PL_i^x * L_i + PW_i^x * W_i) \leq W, \forall i \in \{1, 2, \dots, n\} \quad (5)$$

$$z_i + \mathcal{R}_i * H_i \leq H, \forall i \in \{1, 2, \dots, n\} \quad (6)$$

$$\sum_{i=1}^n \mathcal{R}_i^{yz} = \mathcal{R}_i * (PL_i^x * L_i + PW_i^x * W_i), \forall y \in \{1, 2, \dots, W\}, z \in \{1, 2, \dots, H\}, i \in \{1, 2, \dots, n\} \quad (7)$$

$$\sum_{i=1}^n \mathcal{R}_i^{yz} = \mathcal{R}_i * (PL_i^x * L_i + PW_i^x * W_i), \forall y \in \{1, 2, \dots, W\}, z \in \{1, 2, \dots, H\}, i \in \{1, 2, \dots, n\} \quad (8)$$

$$\sum_{i=1}^n \mathcal{R}_i^{yz} = \mathcal{R}_i * H_i, \forall x \in \{1, 2, \dots, L\}, y \in \{1, 2, \dots, W\}, i \in \{1, 2, \dots, n\} \quad (9)$$

$$PL_i^x = 1 - PW_i^x, \forall i \in \{1, 2, \dots, n\} \quad (10)$$

$$\sum_{i=1}^n (\mathcal{R}_i * V_i) \leq V \quad (11)$$

$$\mathcal{R}_i, \mathcal{R}_i^{yz}, PL_i^x, PW_i^x = 0 \text{ 或 } 1, \forall x \in \{1, 2, \dots, D\}, y \in \{1, 2, \dots, W\}, z \in \{1, 2, \dots, H\}, i \in \{1, 2, \dots, n\} \quad (12)$$

## 3 基于遗传算法建模

遗传算法是一种迭代算法，在初始化过程中会随机产生一组解，后经过模拟进化、继承等遗传操作产生新解，每组解都需要根据评估函数进行评估。

编码方法：算法因子  $S = (M, P, P_x, P_y, T)$  染色体，代表一个解；待装货物的一个排列  $M = (M_1, M_2, \dots, M_n)^T$ ；在排列M状态下的  $\mathcal{R}_i$ 、 $P\mathcal{R}_i^x$ 、 $P\mathcal{R}_i^y$  的排列及装箱物品坐标如下：

$$\begin{cases} P = (\mathcal{R}_1, \mathcal{R}_2, \dots, \mathcal{R}_n)^T \\ P_x = (P\mathcal{R}_1^x, P\mathcal{R}_2^x, \dots, P\mathcal{R}_n^x)^T \\ P_y = (P\mathcal{R}_1^y, P\mathcal{R}_2^y, \dots, P\mathcal{R}_n^y)^T \end{cases} \quad T = (x, y, z) = \begin{bmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ \vdots & \vdots & \vdots \\ x_n & y_n & z_n \end{bmatrix}$$

适应函数：

$$Fitness(S) = \begin{cases} \sum_{i=1}^n B_i V_i / V, & \text{若 } \sum_{i=1}^n B_i V_i \leq V \\ 0, & \text{else} \end{cases} \quad (13)$$

罚函数：通过处理不可行解，将约束问题转化为无约束。估价函数：估计待搜索节点的重要程度。约束条件 (2)，货物存放空间之间不存在交集 (14)；约束条件 (3)，装箱货物不悬空 (15)；约束条件 (4) (5) (6)，货物不超出容器 (16-18)。根据约束条件，要求任何一个都为0，其函数表达式为 (19)；根据算法规则得评估函数为 (20)。

$$g_1(S) = \begin{cases} 0, & \text{若 } \sum_{i=1}^n \mathcal{R}_i^{yz} \leq 1 \\ 1, & \text{else} \end{cases} \quad (14)$$

$$g_2(S) = \begin{cases} 0, & \text{若 } \sum_{i=1}^n \mathcal{R}_i^{yz} - \mathcal{R}_i^{yz(i+1)} \geq 0 \\ 1, & \text{else} \end{cases} \quad (15)$$

$$g_3(S) = \begin{cases} 0, & x_i + \mathcal{R}_i * (PL_i^x * L_i + PW_i^x * W_i) \leq L \\ 1, & \text{else} \end{cases} \quad (16)$$

$$g_4(S) = \begin{cases} 0, & y_i + \mathcal{R}_i * (PL_i^x * L_i + PW_i^x * W_i) \leq W \\ 1, & \text{else} \end{cases} \quad (17)$$

$$g_5(S) = \begin{cases} 0, & z_i + \mathcal{R}_i * H_i \leq H \\ 1, & \text{else} \end{cases} \quad (18)$$

$$G(S) = \sum_{j=1}^5 g_j(S) \quad (19)$$

$$val(S) = Fitness(S) * (5 - G(S)) \quad (20)$$

## 4 数据测试

先将目的地相同的物品混装到包装箱中（如表2所示）。订单信息如表3所示。按需包装后得到表4。

表2 预设包装箱尺寸信息表

名称	长	宽	高	名称	长	宽	高
A1	200	400	400	A4	400	800	800
A2	200	400	800	A5	400	400	400
A3	400	400	800	A6	800	800	500

表3 货物信息表

名称	长	宽	高	数量	名称	长	宽	高	数量
移动硬盘/鼠标	200	100	40	800	HP5200	800	800	250	50
键盘	400	200	25	400	显示器	600	400	150	200
笔记本电脑	600	400	150	200	数码相机	200	200	200	200
台式电脑	800	800	250	200	主机箱	600	400	200	200
HP1566	400	250	250	50	主板	400	400	200	200

表4 预设包装箱明细表

包装箱名	长	宽	高	限重	数量	包装箱名	长	宽	高	限重	数量
A1	200	400	400	7000	125	A4	400	800	800	45000	200
A2	200	400	800	14000	125	A5	400	400	400	18000	200
A3	400	400	800	24000	125	A6	800	800	500	45000	125

根据数据进行仿真测算，得图1装箱方案。根据16车次的装箱计划，可以看到，在第12车次的时候，进行四种货物的混装。在第15车次和第16车次的时候，都还有整块的矩形区域可以利用。

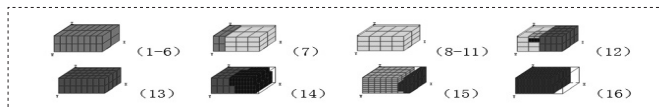


图1 三维装箱图

## 5 结论

遗传算法类似深度搜索，不会因为外界因素而对结果产生影响，具有一定的可靠性。算法主要采用多点搜索方式，在计算执行中并行性高的，提高了计算效率。其选择、交叉、变异等运算都是通过概率计算的方式来进行，使得搜索过程具有极强的灵活特性，可以很好的实现求解的全局优化。在规范化的方面进行了约束，实际操作也会按照规定而变得更加的简单。

### 参考文献

- [1] 罗建军, 吴东辉, 罗细飞. 三维装箱问题的启发式算法[J]. 物流技术, 2012, 31(2): 127-128.
- [2] 张德富, 彭煜, 张丽丽. 求解三维装箱问题的多层启发式搜索算法[J]. 计算机学报, 2013, 35(12): 2553-2561.
- [3] 陈德良, 陈治亚. 三维装箱问题的模型与改进遗传算法[J]. 数学的实践与认识, 2010, 40(2): 142-147.