

一维下料优化的一种新算法

王小东, 李刚, 欧宗瑛*

(大连理工大学 CAD & CG 研究所, 辽宁 大连 116024)

摘要: 针对一维下料优化问题, 提出了一种基于启发式多级序列线性优化思想的新算法, 即将下料优化问题转化为多级序列线性优化问题求解。每级求解时, 在当前可行的下料方式中选择最优的一种进行下料, 不断重复此操作, 直到所有剩余的坯料数目均减小至零为止。原问题的最优解就是各个序列优化问题所求得的最优下料方式的总合。计算表明, 与目前常用的整数线性规划或遗传算法相比较, 该算法有结构简明、计算速度快、节材效果好的优点。

关键词: 一维下料; 整数规划; 遗传算法; 线性优化

中图分类号: TH 123.1 **文献标识码:** A

0 引言

最大限度地节约材料, 提高材料的利用率, 是实际生产中的一个指导原则。一维下料优化问题是讨论从一种规格的材料中, 分切出各种不同长度的坯料, 以使材料的利用率最高。这类优化问题在型材、棒材、管材、金属结构材料、建筑材料, 甚至布料下料中广泛存在。目前, 国内外关于这方面的研究十分活跃, 并涌现出了不少近似算法, 如 Gilmore 与 Gomory 用线性规划建立的一刀切问题的数学模型^[1, 2]; Dyckhoff 提出的线性规划方法^[3, 4] 以及 Sarker 提出的动态规划方法^[5] 等。由于下料问题属于布局 (layout) 问题, 不同于一般的数值性优化, 近年又出现应用遗传算法来求解下料优化问题^[6]。本文在讨论对比目前常用的两种求解方法 (常规整数线性规划方法和遗传算法) 的同时, 提出一种基于启发式多级序列线性优化思想的新方法, 并分别进行算例对比。

1 常规整数线性规划求解方法

1.1 基本思想

穷举出所有可能的下料方式 (cutting pattern), 以所需原材料最少为优化目标, 以每种下料方式的重复次数为优化变量, 按整数线性规划原理求解。

1.2 数学模型

给定 m 种长度的坯料 l_1, l_2, \dots, l_m , 所需的数量分别为 b_1, b_2, \dots, b_m , 已知原材料长度为 L 。

设有 n 种可能下料方式, 每种方式的重复次数为 $x_j (j = 1, 2, 3, \dots, n)$ 。在方式 j 中, 第 i 件坯料的重复次数为 $a_{ij} (i = 1, 2, 3, \dots, m)$, 从而可建立如下数学模型。

目标函数: $\min S = x_1 + x_2 + \dots + x_n$

约束条件:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n & b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n & b_2 \\ \dots\dots\dots \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n & b_m \\ x_1, x_2, \dots, x_n & 0 \end{cases}$$

优化参数变量: x_1, x_2, \dots, x_n 均为非负整数

1.3 所有可能下料方式的计算确定

在上述模型中, 如果所取的下料方式是选择所有可能的下料方式, 则所求得的最优解就是严格全局最优解。所有可能下料方式的确定是决策下料方案的前提, 因此, 人们提出了很多方法, 比较流行的是搜索树 (search tree) 法^[7]。其原理为: 设在同一种原材料 L 上切割 m 种零件毛坯 $l(1) \sim l(m)$ 的个数分别为 $A(1) \sim A(m)$, 则 $l(I)$ 与 $A(J)$ 应该满足下列关系式:

$$(L - \min\{l(1), l(2), \dots, l(m)\}) < \sum_{i=1}^m l(i)A(i)$$

L

即用料小于原材料的长度, 余料小于最短零

件毛坯的长度, 据此确定下料方式 程序流程图如图 1 所示 零件毛坯按长度从大到小排列成 $l(1) \sim l(m)$.

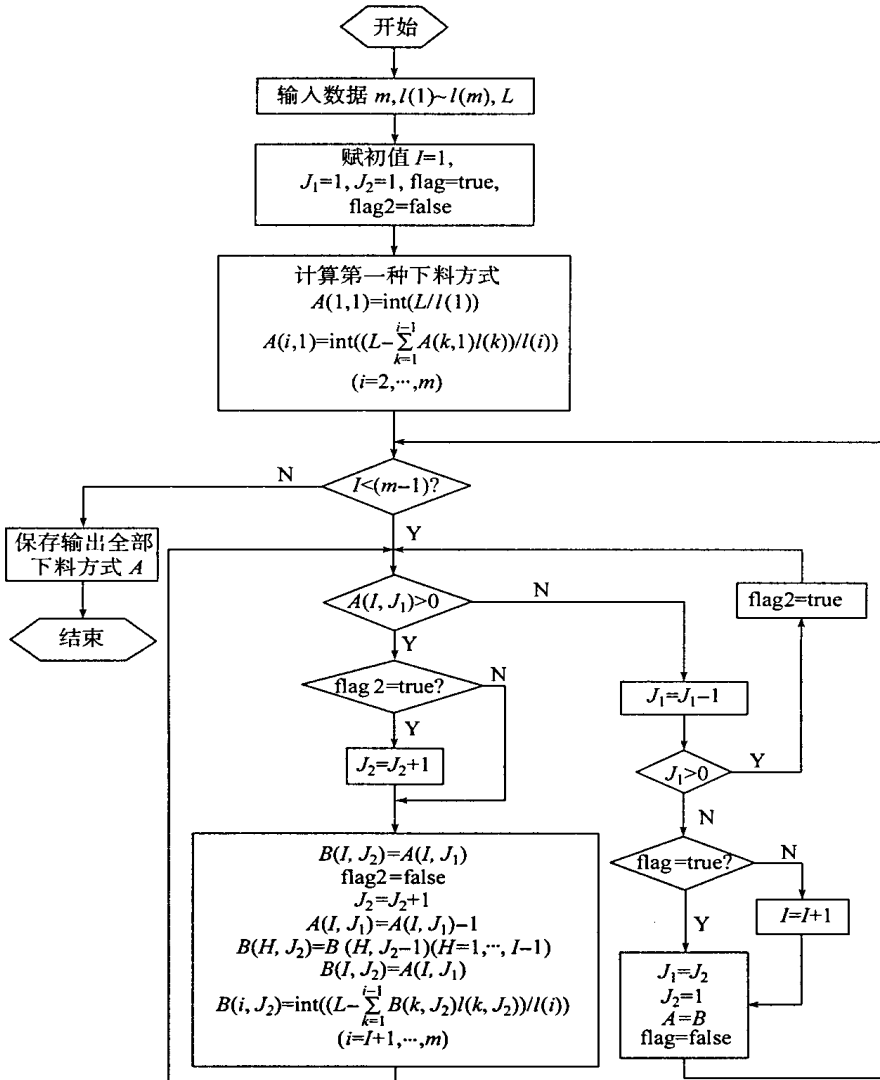


图 1 确定所有可能下料方式的程序流程图

Fig. 1 Flow chart of detemining all possible cutting patterns

例如: 原材料长度 $L = 4\text{ m}$, 需要切割 4 种零件毛坯 ($m = 4$), 长度分别为 38 6 32 2 19 4 和 6 6 cm. 经计算, 所有可能下料方式共有 611 种, 即 $n = 611$.

又如: 原材料长度 $L = 2\text{ 6 m}$, 需要切割 8 种零件毛坯 ($m = 8$), 长度分别为 40 4 36 5 32 4 25 3 20 8 19 4 15 6 和 6 6 cm. 经计算, 所有可能下料方式共有 17 166 种, 即 $n = 17\text{ 166}$

可见, 随着所切割零件毛坯的种类 m 的增大, 当原材料的长度 L 和所切割零件的长度 $l(I)$ 之比大于一定数值后, 所有可能下料方式的总数

将成裂变式增加, n 的增大可能遇到“组合爆炸”问题

1 4 求解方法

由一维下料优化问题的数学模型, 可以看出下料方法是一个整数规划 (integer programming, IP) 问题 传统的求解方法是简单地将整数变量松弛, 作为连续量处理, 简化为常规的线性规划, 采用单纯形方法 (simplex method) 求解, 然后对结果进行调整 但这样做并不能保证得到最优解, 而且调整之后易产生一种或几种零件的过量或不足量, 因此还需要作进一步处理

采用针对求解整数规划的分枝定界(branch and bound)法也只能在变量数目较少时适用(即下料组合方式 n 较小)。由此可见,虽然采用这种方法能求得严格全局最优解,但是当原材料的 L 和所割零件的 $l(i)$ 之比大于一定数值后,就会出现可能的切割下料组合方式很多,即方案数 n 迅速增大问题,使得求解速度缓慢。用分枝定界法求解遇到困难。

2 基于遗传算法的求解方法

该方法的基本思想是把零件的一个顺序作为一种下料方案,并将之视做组合优化问题来求解。先定义解的编码和解码方法以及适应度函数,然后给出初始种群,在求解过程中反复进行交叉(crossover)、变异(mutation)和选择(selection),直到最好解的适应度值达到要求或满足了预定的进化代数,就停止计算,并输出最优解。

遗传算法由于其不成熟性也存在一些不足:首先,在变量多、取值范围大或无给定范围时,收敛速度下降。其次,遗传算法的参数选择尚无定量方法,算法中的交叉率和变异率对它的性能影响很大。最后,遗传算法可找到最优解附近,但无法精确确定最优解位置,即其可以以较快的速度搜索到最优解的90%左右,但要达到真正的最优解则要花费很长时间。一些对比实验表明,如果兼顾收敛速度和解的质量两个指标,单纯的遗传算法未必比其他搜索方法优越^[8]。目前遗传算法用于下料问题的优化多停留在探索研究阶段。

3 启发式多级序列线性优化方法

3.1 基本思想

将下料优化问题转化为多级序列线性优化问题求解。每级求解时,在当前可行的下料方式中选择其中最优化的一种进行下料,并尽可能多地重复使用此种下料方式;然后对剩余的坯料重新优化选取新的当前最优的下料方式,不断地重复上面的操作,直到所有剩余的坯料数目均减小至零为止。原问题的最优解就是各个序列优化问题所求得的最优下料方式的总合。该方法实际是模仿下料操作者实际制定下料方案时常采用的策略。

3.2 当前最优下料方式计算模型

给定 m 种长度的坯料 l_1, l_2, \dots, l_m ,所需的数

量分别为 b_1, b_2, \dots, b_m ,已知原材料长度为 L 。

设在最优一种下料方式中,第 i 件坯料的重复根数为 a_i ,由此建立如下数学模型。

$$\text{目标函数: } \max S = \sum_{i=1}^m a_i l_i$$

约束条件:

$$\begin{cases} 0 & a_i & b_i \\ & \sum_{i=1}^m a_i l_i & L \end{cases} \quad (i = 1, 2, \dots, m)$$

优化参数变量: a_1, a_2, \dots, a_m 均为非负整数。

这是一个整数线性规划问题,由于每次截取的不同长度的坯料种类有限(即该问题中要求的变量个数有限),可以采用分枝定界法来求解。

3.3 多级序列线性优化计算方法

将上述当前最优下料方式计算求解作为多级序列线性优化的子程序,在每级求解中重复调用。

完整的求解过程如下:

Step1 依给定条件调用当前最优下料计算子程序,求解得到优化值 $a_i l_i$ 组成的 $\sum_{i=1}^m a_i l_i$ 作为第一级下料方式。

Step2 计算此种下料方式的重复次数,即此种下料方式所需原材料 L 的根数 d 。

$$d = \min \left(b_1/a_1, b_2/a_2, \dots, b_m/a_m \right)$$

Step3 计算去掉 d 根后,余下的每种待切割的毛坯的根数

$$b_i = b_i - d a_i$$

Step4 将 b_i 作为新一级优化计算的给定值,如果所有的 b_i 都已减小至零,则优化计算结束;否则转Step1,重新调用当前最优下料方式计算子程序,求得新一级的下料方式和重复次数。

Step5 各级最优下料方式及其重复次数的集合即为多级序列线性优化的最终结果。算法流程如图2所示。

4 计算实例

本文中提到的实例均运行在主频为300 MHz,内存为192 MB的计算机上。

4.1 新算法与常规整数线性规划方法的比较

原材料长4 m,需切割的零件坯料分别为长46.3 cm的100件、长40.5 cm的200件、长32.4 cm的200件、长25.6 cm的200件、长18.2 cm的

200 件 求最优下料方案(不考虑切口损失).

用前述常规整数线性规划方法求解, 计算结果如表 1 所示 总计需原材料 70 根, 材料的利用率为 99.89%, 计算用时 76 min.

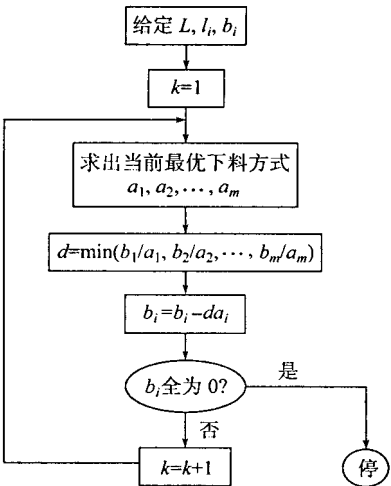


图 2 启发式多级序列线性优化方法程序流程图

Fig.2 Flowchart of heuristic multi-sequential linear optimization algorithm

表 1 常规整数线性规划方法计算结果
Tab.1 Computation results by conventional integer linear programming algorithm

下料 方式	棒 材 类 型					下料 根数
	1	2	3	4	5	
1	5	1	0	5	0	10
2	0	1	3	1	13	1
3	3	1	4	0	5	12
4	0	0	9	2	3	1
5	0	6	0	4	3	27
6	0	2	5	4	3	7
7	3	1	6	1	0	1
8	1	0	9	1	2	11

用本文提出的启发式多级序列线性优化方法求解, 计算结果如表 2 所示 总计需原材料 71 根, 材料利用率为 98.49%, 计算用时 16 s

从该例可以看出, 与常规整数线性规划方法相比, 启发式多级序列线性优化方法虽然不能获得严格全局最优解, 但它大大地节省了时间和空间(计算和存储), 且节材的效果很显著. 而当坯料的类型数增加时, 用常规整数线性规划方法计算需要几个甚至几十个小时, 令人难以忍受. 与

常规整数线性规划方法相比, 新算法材料的利用率稍微有所下降, 但均发生在最后一根; 此例中最后一根剩余 76 cm, 可以为以后切割所用

表 2 启发式多级序列线性优化方法计算结果
Tab.2 Computation results by heuristic multi-sequential linear optimization algorithm

下料 方式	棒 材 类 型					下料 根数
	1	2	3	4	5	
1	5	1	0	5	0	20
2	0	1	4	4	7	25
3	0	3	8	0	1	12
4	0	7	3	0	1	1
5	0	5	1	0	9	1
6	0	9	0	0	1	3
7	0	9	0	0	0	8
8	0	8	0	0	0	1

4.2 新算法与基于遗传算法的求解方法比较
讨论文献[6]中的例子. 原材料长 3 m, 需切割的零件坯料分别为长 2.2 m 的 3 件、长 1.8 m 的 3 件、长 1.2 m 的 4 件、长 0.5 m 的 6 件、长 0.3 m 的 6 件. 求最优下料方案(不考虑切口损失).

用基于遗传算法的求解方法计算, 文献[6]中的结果如表 3 所示

表 3 基于遗传算法的求解方法计算结果(文献[6]例)

Tab.3 Computation results by genetic algorithm (example from Lit [6])

下料 方式	棒 材 类 型					下料 根数
	1	2	3	4	5	
1	1	0	0	0	2	1
2	0	1	1	0	0	2
3	1	0	0	1	1	1
4	0	0	2	0	2	1
5	1	0	0	1	1	1
6	0	1	0	2	0	1
7	0	0	0	2	0	1

用启发式多级序列线性优化方法计算, 结果如表 4 所示. 经对比可知, 两种方法都需要原材料 8 根, 材料的利用率同为 90%, 但采用启发式多级序列线性优化方法, 计算只需 0.5 s. 文献[6]中的例子比较简单, 两种方法的比较不是很明显. 当需要的零件坯料数增加时, 由于基于遗传算法的求解方法对同一问题需要多次运行, 从多种下

料方案中择优, 算法的计算速度将比较慢; 而采用本文提出的新算法, 不但能保证同样高的材料利用率, 而且计算速度有显著提高

表 4 启发式多级序列线性优化方法计算结果 (文献[6] 例)

Tab. 4 Computation results by heuristic multi-sequential linear optimization algorithm (example from Lit [6])

下料 方式	棒 材 类 型					下料 根数
	1	2	3	4	5	
1	0	1	0	0	4	1
2	0	0	0	6	0	1
3	0	1	1	0	0	2
4	1	0	0	0	2	1
5	0	0	2	0	0	1
6	1	0	0	0	0	2

5 结 语

本文模拟了人脑解决问题的过程, 提出了一种基于启发式多级序列线性优化思想的新方法来求解一维下料优化问题。该算法结构简明, 易于实现, 且计算速度快, 节材效果显著。经对 100 组算例进行实验, 结果表明, 材料的利用率普遍达到 93% 以上, 有的甚至超过 99%。由此可见, 综合考虑求解的速度和解的质量, 本文提出的新算法

在工程应用上是求解一维下料问题近似最优解的实用方法

参考文献:

[1] GLMORE P C, GOMORY R E. A linear programming approach to the cutting stock problem (Part I) [J]. **Oper Res**, 1961, **9**: 849-859.

[2] GLMORE P C, GOMORY R E. A linear programming approach to the cutting stock problem (Part II) [J]. **Oper Res**, 1963, **11**: 863-887.

[3] DYCKHOFF H. A new linear programming approach to the cutting stock problem [J]. **Oper Res**, 1981, **29**(6): 1094-1104.

[4] DYCKHOFF H. A typology of cutting and packing problems [J]. **Euro J of Oper Res**, 1990, **44**(2): 145-159.

[5] SARKER B R. An optimum solution for one dimensional slitting problems: A dynamic programming approach [J]. **J Oper Res Soc**, 1988, **39**(8): 749-755.

[6] 贾志新, 殷国富, 胡晓兵, 等. 一维下料方案的遗传算法优化 [J]. **西安交通大学学报**, 2002, **36**(9): 967-970.

[7] 刘勇彪. 等界面长条类材料下料方案的最优化设计 [J]. **机械设计与制造**, 1994, **5**: 12-13.

[8] 陈国良. 遗传算法及其应用[M]. 北京: 人民邮电出版社, 1996.

A new optimization algorithm for one-dimensional cutting-stock problem

WANG Xiao-dong, LI Gang, OU Zong-ying*

(CAD & CG Lab., Dalian Univ. of Technol., Dalian 116024, China)

Abstract: Imitating human intelligence, a new algorithm based on heuristic sequential linear optimization for one-dimensional cutting-stock problem is presented. The main idea of the new algorithm is to process a global optimization problem of the cutting-stock as a sequential optimization problem by multiple stages. During every sequential stage, the best cutting pattern for the current situation is researched and processed. This stage processing is repeated until all the required stocks have been generated. Numerical examples demonstrate that it is advantageous in simplifying the program and elevating computational speed, compared with the conventional methods of linear integer programming or genetic algorithm.

Key words: one-dimensional cutting-stock; integer programming; genetic algorithm; linear optimization