一种基于递归技术的一维下料算法*

A Recursive Algorithm for One-dimensional Cutting Stock Problem

郑 文[1],崔耀东[2],周 密[2]
ZHENG Wen[1], CUI Yao-dong[2], ZHOU Mi[2]
(广西大学计算机与电子信息学院,广西南宁 530004)
(School of Computer, Electronics and Information, Guangxi University, Nanning, Guangxi, 530004, China)

摘要:用递归技术进行排样,并将排样方式与线性规划相结合,提出一种基于递归技术的一维下料算法. 该算法通过约束一个排样方式中所含毛坯种数,达到减少开堆数的目的,利用上界技术来减少计算时间. 该算法可以大幅缩短计算时间,在材料利用率基本不下降的情况下,可以明显减少最大开堆数.

关键词:递归算法 一维下料 开堆 线性规划
中图法分类号:TP391.72 **文献标识码**:A **文章编号**:1002-7378(2010)04-0415-03

Abstract: A recursive algorithm to solve one-dimensional cutting stock problem is proposed. A recursive procedure is used for pattern generation and articulated with linear programming to generate the cutting plan. The algorithm reduces the open stacks by means of constraining the number of item types in each pattern, and using branch and bound method to decrease computing time. The computational results indicate that the presented approach can decrease the computing time significanthy and reduce effectively the open stacks with high material utilization rate.

Key words: recursive algorithm, one-dimensional cutting stock, open stacks, linear programming

一维下料问题[1]是指用长 L 的线材(如型材、棒材和管材等)切出 m 种毛坯,第 i 种毛坯的长度为 l_i,需求量为 b_i($i=1,\cdots,m$),当排样的目标满足全部毛坯的需求时,使所需的线材根数最小.

在线材切割过程中,切下的毛坯在下料机床周围按堆存放,每堆中只含一种(长度相同的)毛坯[2]. 每切下一种新的毛坯,一个新的堆就会被打开. 当该种毛坯全部切出后,相应的堆才会关闭,腾出场地给其它堆. 处于打开状态的堆称为开堆. 设下料方案由 n 个按切割顺序编号的排样方式组成,如果含第 i 种毛坯的所有排样方式中,序号最小的为 k_1,最大的为 k_2($1\leqslant k_1\leqslant k_2\leqslant n$),则相应的堆在开始处理排样方式 k_1 时打开,在处理完排样方式 k_2 时关闭. 下料过程中不同时刻,开堆数可能不同,其最大值称为最大开堆数. 因下料场地面积一定,(最大)开堆数往往会受限制.

人们通常采用近似算法,对下料方案中的排样方式进行排序以减少开堆数. 文献[3]指出通过减少每个排样方式中所含毛坯种数可以减少开堆数. 本文提出一种基于递归技术的排样方式,通过约束一个排样方式中所含毛坯种数,达到减少开堆数的目的,利用上界技术减少计算时间,再将递归排样方式与线性规划相结合,生成一维下料方案算法. 实验结果表明,该方法可以有效减少开堆数.

1 递归排样方式

下料算法由多个排样方式组成,为生成当前排样方式 P,需要求解整数规划问题:

收稿日期:2010-08-15
作者简介:郑 文(1985-),男,硕士研究生,主要从事优化计算技术与CAD研究。
*国家自然科学基金项目(61063031),广西大学科研基金项目(XB2100145)资助。

$$V_{\max}=\max\ \sum_{i=1}^{m}v_iy_i;\sum_{i=1}^{m}l_iy_i\leqslant L;\sum_{i=1}^{m}\delta(y_i)\leqslant K;y_i\text{ 非负整数},i=1,\cdots,m. \quad (1)$$

其中，$V_{\max}$ 为当前排样方式的最大价值，v_i 为第 i 种毛坯的当前单价，y_i 为排样方式中含第 i 种毛坯的个数，K 为最大毛坯种数约束，即单个排样方式中所含毛坯的种数，不能超过 K，当 $y_i>0$ 时，$\delta(y_i)=1$，否则 $\delta(y_i)=0$. 设 $F(x)$ 为不考虑最大毛坯种数约束时，长为 $x(0\leqslant x\leqslant L)$ 的线材的最大价值. 采用动态规划算法求解背包问题[4] 来确定 $F(x)$.

$$F(x)=\max\ \sum_{i=1}^{m}v_iy_i;\sum_{i=1}^{m}l_iy_i\leqslant x;y_i\text{ 非负整数},i=1,\cdots,m.$$

采用递归技术求解问题(1)，每次调用递归函数 $G(x,V_0,K_0,Y)$ 都从剩余线材 x 上切下一个毛坯，这里 x 为剩余线材长度，V_0 为已切下毛坯的总价值，K_0 为已切下毛坯的种数，Y 为有 m 个元素的数组，其第 i 个元素 Y_i 表示已切下第 i 种毛坯的个数，$V_0=v_1Y_1+\cdots+v_mY_m$. 设 $l_{\min}$ 为最短毛坯的长度，V 为剩余线材 x 中将排入的毛坯所能达到的最大价值. 初始令 $V_{\max}=0$. 递归排样如下：

1. 若 $x<l_{\min}$ 且 $V_0>V_{\max}$，令 $V_{\max}=V_0$，$y_i=Y_i(i=1,\cdots,m)$，返回零；

2. 令 $V=0$；

3. For $i=1$ to m；

4. 若 $v_i\leqslant 0$ 或 $l_i>x$，跳过第 i 种毛坯；

5. 令 $x_1=x-l_i$；若 $v_i+F(x_1)\leqslant V$，跳过第 i 种毛坯；

6. 若 $K_0=K$ 且 $Y_i=0$，跳过第 i 种毛坯；

7. 若 $Y_i=0$，令 $K_1=K_0+1$，否则令 $K_1=K_0$；

8. 令 $Y_i=Y_i+1$，$V_1=V_0+v_i$，$U=v_i+G(x_1,V_1,K_1,Y)$，$Y_i=Y_i-1$；

9. 若 $U>V$，令 $V=U$；

10. End For；

11. 返回 V.

1 行表示剩余线材 x 太小，已经不能排入任何毛坯的情况. 这时如果所对应的排样方式优于当前最好排样方式 ($V_0>V_{\max}$)，就将当前排样方式记为当前最好排样方式. 3～10 行表示从 m 种毛坯中选择切下一个毛坯. 如果当前毛坯价值非正，或者其长度超过 x，则不得排入该种毛坯(4 行). 如果排入当前毛坯，则对应分支的价值上界为 $v_i+F(x_1)$，如果该上界小于等于剩余线材 x 的当前价值 V，则不选择当前毛坯(5 行). 当 $Y_i=0$ 时，当前毛坯将作为一种新毛坯出现在排样方式中，这时如果排入的毛坯种数 K_0 已经达到所允许的最大值 K，则不能排入当前毛坯(6 行). 7 行和 8 行指出，排入当前毛坯后，所对应分支的价值 U，等于当前毛坯的价值 v_i 和剩余线材 x_1 的价值 $G(x_1,V_1,K_1,Y)$ 之和，并且在调用递归函数前后对向量 Y 进行了修正. 9 行指出，当 $U>V$ 时，应该更新剩余线材 x 的价值. 11 行表示返回剩余线材 x 的价值.

2 下料算法

下料问题的决策变量是每个排样方式的使用次数，取非负整数值. 将整数约束去掉后，得到线性规划(LP)问题[5]：

$$\min z=CX;AX\geqslant B,X\geqslant 0,$$

其中，A 是 m 行 m 列的矩阵，其元素 a_{ij} 表示第 j 种排样方式中含第 i 种毛坯的数量，$1\leqslant i,j\leqslant m$；$A$ 的每一列表示一个排样方式；$B=[b_1,\cdots,b_m]$；C 是 m 个元素的行向量，每个元素值都是 1. $X=[x_1,\cdots,x_m]^{\mathrm{T}}$，$x_j$ 为排样方式 j 的使用次数，$1\leqslant j\leqslant m$.

用单纯形法求得 LP 问题的最优解时，决策变量的值一般不是整数，需要采用适当的方法取整，得到下料问题的近优解.

下料算法的步骤如下：

步骤 1　初始取 A 为单位矩阵，表示第 i 个排样方式含第 i 种毛坯 1 个，不含其它毛坯，其使用次数为 b_i 次. 初始解的目标函数值(消耗线材的根数)为 $z=\sum_{i=1}^{m}b_i$.

步骤 2　令毛坯当前的价值向量 $V=[v_1,\cdots,v_m]=CA^{-1}$，按递归排样方式确定当前排样方式 P 及其价值 $V_{\max}$.

步骤 3　如果 $V_{\max}$ 大于 1，根据单纯形法规则，用 P 替换矩阵 A 的一列，从而得到一个新解，转步骤 2；否则转步骤 4.

步骤 4　采用如下取整方法对 LP 的最优解进行取整，得到近优解：将向量 X 的元素都向上取整，所有毛坯的需求一定能满足，但部分毛坯会多余；然后逐个考察各排样方式，在满足毛坯需求的前提下，最大限度地减少各个排样方式的使用次数，使多余毛坯的总量减少.

3 实例验证

用 C＋＋实现算法，使用的计算机主频

1.66GHZ,内存 1.5GB. 选择文献[6]中的第 16 组例题中的 10 道例题(前 11 题排除第 8 题,因其毛坯种数较少),每题 20 种毛坯. 考虑到原例题的线材长度较小并且 LP 法适合于毛坯需求量较大的情况,将原例题所有毛坯的需求量都扩大 10 倍,取定线材长度为 3500mm.

表 1 结果显示,最大开堆数 N 随着 K 的减小而减少;当 K 分别取 2,3 和无穷大时,N 的平均值分别为 9.6,13.2 和 16.5.

表 1 最大开堆数实验结果(30 个排样方案)

K	N										
	例 1	例 2	例 3	例 4	例 5	例 6	例 7	例 8	例 9	例 10	平均值
2	9	10	9	12	10	8	8	13	9	8	9.6
3	16	11	15	11	12	13	13	15	14	12	13.2
$+\infty$	15	17	16	13	17	18	17	18	16	18	16.5

表 2 所示为各排样方案的材料利用率数据,其中 U^* 为 LP 最优解对应的材料利用率,U 为取整后所得近优解的材料利用率.

表 2 各种排样方案的材料利用率数据

题号	$K=2(\%)$		$K=3(\%)$		$K=+\infty(\%)$	
	U	U^*	U	U^*	U	U^*
1	99.65	99.78	99.84	99.99	99.88	100.00
2	99.46	99.65	99.79	99.97	99.79	100.00
3	99.14	99.24	99.86	99.95	99.94	99.99
4	99.67	99.81	99.86	100.00	99.82	100.00
5	99.70	99.84	99.87	99.98	99.90	100.00
6	99.52	99.70	99.84	99.98	99.89	100.00
7	99.46	99.63	99.85	99.99	99.85	100.00
8	99.57	99.72	99.92	99.99	99.82	100.00
9	99.61	99.75	99.84	99.99	99.95	100.00
10	99.57	99.71	99.88	99.99	99.91	100.00
平均	99.54	99.68	99.86	99.98	99.88	100.00

由表 2 结果可以看出,$K=3$ 和 K 无约束的情况相比,排样方案的平均材料利用率接近(U 值分别为 99.86% 和 99.88%,U^* 值分别为 99.98% 和 100%),但最大开堆数有较大差别(前者 13.2,后者 16.5). 这说明通过选择适当的 K 值,可以在材料利用率基本不下降的情况下,明显地减少最大开堆数.

表 3 结果表明,当 $K=3$ 时,采用上界法(参见第 1 节递归排样的第 5 行)的平均计算时间,仅为不采用上界法平均计算时间的 18.6%. 生成排样方式时利用上界法可以大幅缩短计算时间.

表 3 计算时间的实验结果 *

题号	$K=2$(s)		$K=3$(s)	
	t_1	t_2	t_1	t_2
1	1.50	2.59	35.1	169.4
2	4.42	9.83	839.5	>3600
3	1.33	4.92	29.8	652.7
4	3.09	5.99	135.9	1053.8
5	1.86	4.27	119.6	870.5
6	1.13	2.89	51.1	518.1
7	5.88	13.7	564.7	>3600
8	7.31	16.8	629.0	>3600
9	1.39	2.50	68.0	379.8
10	2.41	7.03	377.2	2958.5
平均	3.0	7.1	285.0	>1533.6

* t_1 和 t_2 分别表示采用和不采用上界法时的平均计算时间,>3600 表示计算时间超过 1h,程序被强制中止.

4 结束语

本文采用基于递归技术的排样方式生成算法,约束一个排样方式中所含毛坯种数. 结果表明,该算法可以在基本不引起材料利用率下降的前提下,明显减少开堆数,从而减少下料场地所需面积. 算法设计简单,易于理解,有一定的参考价值,但是递归技术是一种精确算法,计算时间较长. 近似算法,在效果接近精确算法的前提下,能够快速计算,这可作为今后的研究目标.

参考文献:

[1] 崔耀东. 计算机排样技术及应用[M]. 北京:机械工业出版社,2004:13.
[2] Yanasse H H, Lamosa M P J. An integrated cutting stock and sequencing problem[J]. European Journal of the Operational Research, 2007, 183:1353-1370.
[3] Armbruster M. A solution procedure for a pattern sequencing problem as part of a one-dimensional cutting stock problem in the steel industry[J]. European Journal of the Operational Research, 2002, 141:328-340.
[4] Kellerer H, Pferschy U, Pisinger D. Knapsack problems[M]. Berlin: Springer, 2004:192-194.
[5] Cui Y, Chen Y, Wu J. Selecting the best sheet length for the steel stock used in circular blank production [J]. IIE Transactions, 2006, 38 (10): 829-836.
[6] Umetani S, Yagiura M, Ibaraki T. One-dimensional cutting stock problem to minimize the number of different patterns [J]. European Journal of the Operational Research, 2003, 146:388-402.

(责任编辑:尹 闯)