

文章编号:1000-582X(2005)03-0057-04

基于启发式搜索和背包算法的分布式排样系统

魏群义,彭晓东,尹爱军
(重庆大学机械工程学院,重庆 400030)

摘要:针对理论上属于 NP 完全问题的二维矩形件优化排样问题,构建了一个排样效率高、计算速度快和排样效果好的一种近似算法。并结合计算机网络的大量应用,将算法应用于分布式排样系统。算法的主要思想是采用启发式搜索和背包算法,使每次排样都达到局部最优,从而得到近似最优解。并与其它 2 种近似算法进行了比较,指出了它们的不足之处,还给出了一个排样实例。

关键词:矩形件优化排样;启发式搜索;背包算法;分布式系统

中图分类号:TP311.1

文献标识码:A

矩形件排样优化是指给定一定数量的大小各异的矩形件和一定大小的板材,要求将矩形件整齐地排列在板材上,使所需要的板材数量最少。这项工作广泛用于机械制造、轻工、家具以及玻璃切割等行业。矩形件优化排样属于典型的组合优化问题,具有最高的计算复杂性,属于 NP 完全问题。也就是说即使使用当今最快的计算机在人们可接受的时间内也不可能求出这种问题的最优解。而由于生产的需要,人们又迫切需要科技工作者对这一问题给出一个能满足生产需要的解决方法。这些方法尽管得不到最优解,但是能以较快的速度给出一个能满足实际生产需要的近似最优解。国内外学者在这方面作了大量工作,构建了一些近似算法^[1-6]。笔者提出了一种近似算法,采用启发式搜索和背包算法,使每次排样都达到局部最优,从而得到近似最优解,算法既保证了排样利用率,又保证了排样速度。

随着计算机网络的普及应用,实现基于计算机网络的分布式排样系统的需求变得越来越强烈,飞速发展的计算机技术也使其实现成为可能。

1 分布式排样系统的实现

1.1 系统结构模式

分布式排样系统采用 C/S 模式。C/S 模式的最大特点在于将应用程序的数据层与展示层分离,让应用程序的展现归展现,数据归数据,数据放在后端的数据

库服务器,前端 PC 负责数据的展现接口,可以真正达到多人数据共享的目的,解决单击应用程序的困扰。C/S 模式还可以修改应用程序的设置来满足客户的不同需求,做到区别对待。

1.2 系统架构及功能

系统框架如图 1 所示。图中中心数据库指数据库服务器,它存储着板材仓库中所有的板材信息;双向箭头指计算机网络;加工中心指分布在各地的加工中心,它有本地数据库,存储着本地仓库的板材信息(可能包括本地加工中心每次排样剩下的余料)。系统工作原理描述如下:加工中心将板材加工任务信息输入到本地数据库,对程序进行相应设定,如板材运输费用、废料尺寸等,运行系统排样模块,系统将结合本地板材库和中心板材库,得到一个近似最优排样方案,排样效率高,运输费用低。

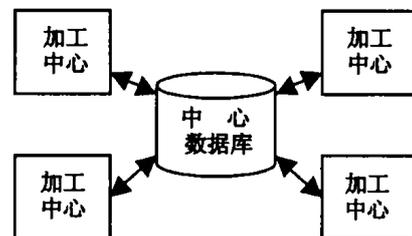


图 1 分布式排样系统架构

2 排样算法

排样的一些基本原则有:优先保证大件(即长度大、小板材宽度的矩形件),因为优先排小件可能造成

无法排下大件;为了便于加工,应尽量将同种矩形件排列在一起,这也是提出条料(条料就是在板材横向和纵向各有一个或者一个以上的同种矩形件的集合)的原因;采用一刀切,因为切割过程中不允许转角,还有不希望退刀,退刀将降低切割效率。

排样方式有竖排和横排 2 种,切割方式也有竖切和横切 2 种。因此排样模式共有 4 种,分别为横排竖切、横排横切、竖排竖切和竖排横切。

2.1 初排算法

设板材的长为 L , 宽为 $W (L \geq W)$ 。所有待排矩形件共有 k 种, 长度为 l_i , 宽度为 $w_i (1 \leq i \leq k)$ 。

在初排算法中, 采用启发式搜索, 启发的原则是使条料排下后剩下的余料 A 的面积最小, 据此来确定最佳排样矩形件及排样方式。算法中首先判断未排件中是否还有大件, 如果有, 则进入大矩形件选择算法; 如果没有大矩形件, 则进入小矩形件选择算法。

2.1.1 大矩形件选择算法

大件只要 2 种排样方式: 横排竖切和横排横切 (如图 2 所示)。对于给定的大件 i , 分别计算 2 种排样方式下余料 A 的面积, 求出较小的面积。即计算下式的最小值:

$$\min \{ (W - w_i y) l_i x, (L - l_i x) w_i y \} \quad (1)$$

式中: y 为板材纵向上矩形件 i 的排列个数;

x 为板材横向上矩形件 i 的排列个数;

n_i 为未排矩形件 i 的个数; 且 $xy \leq n_i, W - w_i y \geq 0, L - l_i x \geq 0$ 。

计算所有未排大件的式(1)最小值, 从中选择最小的一个作为本次排样的矩形件, 确定排样方式, 产生一个条料, 其效果如图 2。

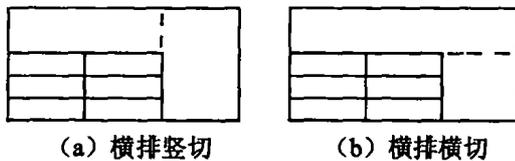


图 2 大矩形件的 2 种排样效果图

2.1.2 小矩形件选择算法

小矩形件有 4 种排样方式: 横排竖切、横排横切、竖排竖切和竖排横切。对于给定的小件 i , 分别计算四种排样方式下余料 A 的面积, 求出最小值。式(2)计算的是小件横排时余料 A 面积的最小值, 式(3)计算的是小件竖排时余料 A 面积的最小值。

$$\min \{ (W - w_i y) l_i x, (L - l_i x) w_i y \} \quad (2)$$

式中: x, y 和 n_i 的意义同式(1); 且 $xy \leq n_i, W - w_i y \geq 0, L - l_i x \geq 0$ 。

$$\min \{ (W - l_i y) w_i x, (L - w_i x) l_i y \} \quad (3)$$

式中: x, y 和 n_i 的意义同式(1); 且 $xy \leq n_i, W - l_i y \geq 0, L - w_i x \geq 0$ 。

取式(2)和式(3)的较小值作为矩形件 i 排列时余料 A 面积的最小值, 计算所有未排小件排列时余料 A 面积最小值, 从中选择余料 A 面积最小的一个矩形件作为本次排样的矩形件, 确定排样方式, 产生一个条料, 其效果见图 3。

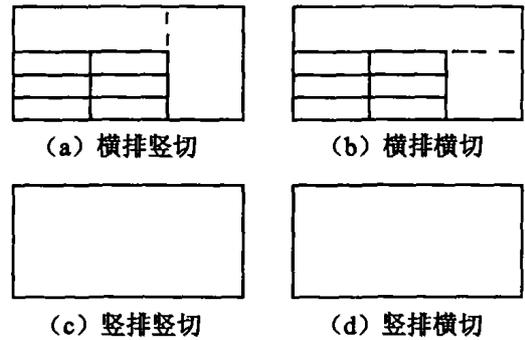


图 3 小矩形件的 4 种排样效果图

2.2 余料 A 的排样方案

经过初排算法后, 余料 A 一般来说是一块面积较小的矩形或者在一个方向上尺寸较小, 如果继续采用前面的初排算法, 将使排样结果成锯齿性, 排样效果不好, 也不利于加工。在这里, 优先考虑另一种排样方案。在这个方案中, 只考虑沿余料 A 长度方向切割, 而且长度方向上只排一种矩形件。如果矩形件 i 横排 $m (m \leq n_i)$ 个, 且 $LIMITSPACE \geq l_a - ml_i \geq 0, l_a$ 为余料 A 的长度, $LIMITSPACE$ 为预先设定的值, 系统中将其设为废料长度和宽度之和的一半, 则找到了填充余料 A 的一种组合, 组合的个数为 n_i/m (取整数); 如果矩形件 i 纵排 $m (m \leq n_i)$ 个, 且 $LIMITSPACE \geq w_a - mw_i \geq 0, w_a$ 为余料 A 的宽度, 则找到了填充余料 A 的一种组合, 组合的个数为 n_i/m (取整数)。这种组合排样利用率高, 利于加工。如果所有组合在余料 A 宽度方向的尺寸之和大于余料 A 的宽度, 就调用背包算法, 得到最优组合, 得到效果如图 4; 反之, 就不调用背包算法, 采用前面的初排算法。

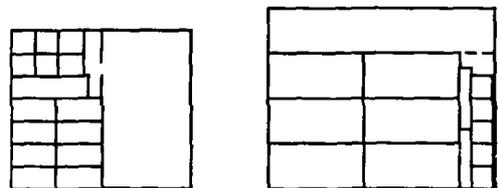


图 4 对余料 A 采用背包算法后的排样效果图

这种优先排样方案成功了将二维排样转化为一维问题, 不仅简化了算法, 而且使排样效果良好, 便于加工。

2.3 背包算法

设 w_i 为 m 个组合在余料 A 宽度方向的尺寸, w_a 为

余料 A 的宽度,选取其中一些组合,使其 $\sum w_i$ 最大但不超过 w_a 。数学模型为:

$$\max \sum x_i w_i \quad 1 \leq i \leq m$$

约束条件: $\sum x_i w_i \leq w_a (1 \leq i \leq m), \sum w_i \geq w_a, x_i = 0$ 或 1。

这是一个 0-1 规划问题。笔者采用改进的动态规划法^[7]求解。

2.4 整块板材排样算法描述

因为每次初排过后,都会产生两块余料,随着排样的继续,余料会越来越多,所以构建了一个余料链表来存储剩下的余料。

整块板材的排样算法简单描述如下:

- 1) 若有未排零件,采用初排算法,排下一块条料,产生余料 A 和 B;如果没有未排零件,排样结束,返回。
- 2) 将临时余料 B 添加到临时余料链表。
- 3) 如果 A 排不下矩形件,转到 4);否则,采用余料 A 的排样方案对余料 A 进行排样。
- 4) 如果临时余料链表中无余料,返回;否则,取一块临时余料 B,如果 B 可以排下矩形件,将 B 作为原板,转到(1),此时 L 和 W 分别为余料 B 的长度和宽度;如果 B 排不下矩形件,转到 4)。

3 与其它 2 种近似算法的比较

文献[4]的近似算法中也采用了背包算法,与本文不同的是,它是对原板材采用背包算法。这就产生一个潜在的问题,如果原板材尺寸较大,那么得到的组合数将会很多。这将影响排样速度,因为随着组合数的增加,背包算法所需的时间呈指数级增长。笔者算法中先进行初排,再对余料 A 采用背包算法。余料 A 的尺寸小于原板材的尺寸,这将大大减少了组合的数量,从而减少了背包算法所耗费的时间,提高了排样速度。

文献[5]的近似算法中也采用了启发式搜索,与本文不同的是,它的启发原则是余料 A 在板材切割方向的尺寸最小,这种算法比较简便。但可能余料 A 在板材切割方向上的尺寸较小,而另一个方向的尺寸很大,这就不一定是最佳排样矩形件。笔者算法采用的启发的原则是使条料排下后剩下的余料 A 的面积最小,排样利用率更高,更接近于最优解。

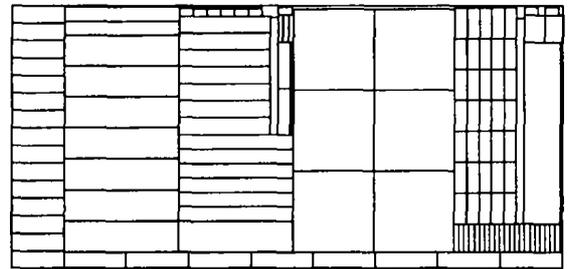
4 计算实例

根据以上算法,笔者使用 Visual C++ 6.0 开发了一个矩形件排样系统。对若干实际问题进行了排

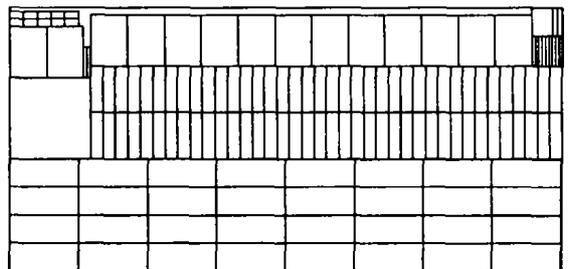
样,效果良好,这里给出其中 1 个问题的数据和排样效果图。数据见表 1^[4](表中长宽的单位均为 mm, i 表示待排样板材编号, n_i 表示板材 i 的数量, l_i 表示板材 i 的长度, w_i 表示板材 i 的宽度),排样板材的长为 12 030 mm,宽为 2 550 mm。排样利用率为 93.92%。其排样效果见图 5。

表 1 待排板材数据表

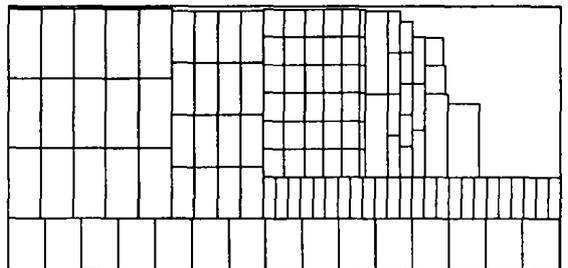
i	n_i	l_i	w_i	i	n_i	l_i	w_i
1	96	450	270	9	8	1 993	165
2	48	400	270	10	16	710	675
3	16	1 140	170	11	16	500	500
4	8	2 500	300	12	32	800	500
5	8	2 500	139	13	30	300	74
6	40	300	270	14	8	1 360	160
7	32	1 500	270	15	8	1 760	785
8	32	270	110				



(a) 第 1 块板材排样结果



(b) 第 2 块板材排样结果



(c) 第 3 块板材排样结果

图 5 板材排样效果图

5 结束语

笔者对若干组数据(包括大数据量)进行了测试,系统的计算时间均不超过 10 s,而且排样利用率均高于 90%。这说明这种近似算法计算速度快,排样效果好,排样利用率高,完全可以满足生产实际需要,具有很好的推广价值。

在算法设计及系统实现、测试过程中,得到了重庆大学硕士研究生王远斌、张帆和王竞的大力帮助,在此表示感谢。

参考文献:

- [1] CHRISTOFIDES N. An Algorithm for Two-dimensional Cutting Problems [J]. *Operations Research*, 1977, 25 (1): 30 - 40.
- [2] WANG P T. Two Algorithms for Constrained Two-dimensional Cuttings Stock Problems[J]. *Operations Research*, 1983, 31(3):573 - 586.
- [3] RHEE W T. Optimal Bin Packing with Items of Random Sizes[J]. *Mathematics of Operations Research*, 1988, 13(1): 140 - 151.
- [4] 曹炬,周济. 矩形件排样优化的背包算法[J]. *中国机械工程*, 1994, 5(2):11 - 12.
- [5] 曹炬,周济. 矩形件排样优化的一种近似算法[J]. *计算机辅助设计与图形学报*, 1995, 7(3):190 - 195.
- [6] 文贵华,丁月华. 基于动态分割与合一的排样算法[J]. *计算机工程*, 1999, 25(12):23 - 24.
- [7] 王晓东. 计算机算法设计与分析[M]. 北京:电子工业出版社, 2001.

Distributed Cutting Stock System Based on Heuristic Search and Knapsack Algorithm

WEI Qun-yi, PENG Xiao-dong, YIN Ai-jun

(College of Mechanical Engineering, Chongqing University, Chongqing 400030, China)

Abstract: A approximate algorithm is presented by which satisfactory results of two dimensional optimal layout belonged to NP complete problems can be obtained efficiently and rapidly. The algorithm integrated with widespread of computer network has been applied to the distributed cutting stock system Adopting Heuristic search and Knapsack algorithm, the algorithm makes each layout optimal locally and obtained the approximate optimal solution. It is compared with other two approximate algorithms and pointed out their shortcomings. In the end of the article a layout example was presented.

Key words: orthogon optimal layout; heuristic search; knapsack algorithm; distributed system

(编辑 成孝义)