⑰
92-97

定长条材优化下料的实用算法研究*

TH140.8

A Study on the Practical Algorithm for One-Dimensional Equal Stock Cutting Problem

龚　坚　　刘　飞　　徐宗俊
Gong Jian　　Liu Fei　　Xu Zhongjun
（重庆大学机械工程一系，重庆，630044；第一作者 28 岁，男，博士生）

A **摘　要**　在分析定长条材优化下料数学模型基础上，构造了一个背包列生成近似算法，从工程实践角度给出了此问题的求解方法，并给出了计算实例。

关键词　优化下料问题；最优化；背包列生成
中国图书资料分类法分类号　TP21

条材　　定长条材．下料．数学模型

ABSTRACT　Based on alalysing the mathematical models for one-dimensional equal stock cutting problem, a knapsack column generation algorithm has been constructed, which is a successive approximate algorithm for the practical engineering. In the end of this paper, some computational examples have been presented, which show that the convergence and optimum results of the algorithm are quite good.

KEYWORDS　cutting-stock problem; optimization; knapsack column generation

0 引　言

随着全球对资源消耗问题的日益重视及降低消耗也是环境问题治本措施等新观念的出现，制造系统资源优化利用的研究已愈来愈重要。制造过程中的优化下料问题就是制造系统中资源优化利用问题之一。优化下料涉及的问题很多，其中定长条材优化下料问题是制造过程中常遇到的下料问题。钢材、铝材合金型等的切割中许多都是定长条材下料问题。我国机械制造、建筑等行业每年消耗的钢材和铝合型材数量巨大，涉及大量的定长条材下料问题。定长条材下料问题一般描述为：在一定数量的给定长度的原材上，如何以最少的原材根数切割出所需数量的各种长度的零件。目前，一般的人工排料法是在给定的原材上先切割出长度最大的零件，再在剩下的零件中切割出可能的最长零件，依次切割，直至最后无法节割出任何一种规格零件为止，剩下的边角余料就当废料处理。这种人工排料方法除浪费十分惊人外，排料计算复杂、耗费工程技术人员的大量精力，且容易排料出错。从计算复杂性理论上看，优化下料问题属 NP－完全问题，即不存在多项式界算法。计算机科学家认为求解一个问题的解法，仅当其复杂性随输入规模的增加而多项式增长时，这个算法才是实际有效的。因

* 收文日期　1996-01-11

此，对于 NP—完全问题，通常只能用那种虽然不能保证是最优解但是接近最优解的近似算法来求解。为此，国内外关于这方面的研究十分活跃，并提出了不少近似算法[1~7]，如闵仲求[5]提出的子优下料方法，利用组合优选方法来避免对初始切割方式的枚举，但是，这种方法对于零件种类较多时应用困难，HARLD DYCKHOFF[3]提出的线性规划方法以及 B. R. SARKER[4]提出的动态规划方法等，算法复杂，未能很有效地解决巨大数量初始切割方式的优选问题。笔者针对下料模型的特点，构造了一个背包列生成算法，从工程实践的角度，成功地给出下料问题近似最优解的求解方法。

1　数学模型的选择

设原材长为 L，待下的不同长度的零件种数为 m，第 i 种零件的长度为 l_i，需要量为 b_i，则切割方式的含义为：在长为 L 的条材上同时下长为 l_1 的零件 a_1 个，长为 l_2 的零件 a_2 个，…、下长为 l_i 的零件 a_i 个，$a_i \in \left[0, \min\left\{b_i, \mathrm{int}\left(\frac{L}{l_i}\right)\right\}\right]$，且为整数，显然下料方式不是唯一的，用 j 表示第 j 个切割方式，则 a_{ij} 表示第 j 个切割方式切割出的第 i 种零件的数量，如令 δ_j 表示第 j 个切割方式留下的废料头，n 表示下料方式的总数，则 $\delta_j = L - \sum_{i=1}^{m} a_{ij} \cdot l_i$，$\delta_j \geqslant 0$，一般的下料模型以总废料最少为目标函数，有：

$$\min Z = \sum_{j=1}^{n} \delta_j \cdot x_j$$

$$\text{s. t.} \quad \sum_{j=1}^{n} a_{ij} \cdot x_j \geqslant b_i, \quad x_j \geqslant 0, j = 1, \cdots, n; i = 1, \cdots, m$$

笔者早期的研究[1]已证明该模型在处理实际下料问题时是有一定问题的，在某些情况下，其最优解的原材料利用率很低，甚至模型可能无解。为此，笔者选择下面以使用原材根数最少为目标函数的数学模型(ILP)

$$\min Z = \sum_{j=1}^{n} x_j$$

$$\text{s. t.} \quad \sum_{j=1}^{n} a_{ij} \cdot x_j \geqslant b_i, \quad x_j \geqslant 0 \text{ 且为整数}, j = 1, \cdots, n; i = 1, \cdots, m$$

x_j 为第 j 种切割方式的使用次数。

2　模型的列生成算法

由上节可见，下料问题(ILP)是一个大型整数线性规划问题，要直接求解上述模型是十分困难的。首先，模型中的切割方式列向量是未知的，切割方式总数 n 很大，也即(ILP)规划的决策变量个数相当多。例如，在 6 m 长的条材上，套裁 40 种尺寸从 60 mm 到 240 mm 的零件，可行的切割方式总数很容易超过一千万或一亿个[2]，即使是 4 种零件套裁，切割方式个

数也很容易超过一千。要求解如此多决策变量的线性规划问题，即使用当今的高速计算机来求解也是十分困难的。为此，笔者提出下面的近似算法，首先考虑不带整数约束的下料模型（记为 LP 模型）

$$\min Z = \sum_{j=1}^{n} x_j \tag{1}$$

$$\text{s.t.} \quad \sum_{j=1}^{n} a_{ij} \cdot x_j \geqslant b_i, \quad x_j \geqslant 0, j = 1,\cdots,n; i = 1,\cdots,m \tag{2}$$

2.1　列生成算法的原理

（2）式加上松弛变量可写成：

$$y_i - \sum_{j=1}^{n} a_{ij} \cdot x_j = -b_i, \quad y_i \geqslant 0, i = 1,\cdots,m \tag{3}$$

令 $\boldsymbol{Y} = (y_1, y_2, \cdots, y_m)^T$，$\boldsymbol{X} = (x_1, x_2, \cdots, x_n)^T$，$\boldsymbol{A} = (a_{ij})_{m\times n}$，$\bar{\boldsymbol{b}} = (-b_1, -b_2, \cdots, -b_m)^T$，则(3)式记为：

$$(I \ \vdots \ -A) \cdot \begin{pmatrix} \boldsymbol{Y} \\ \boldsymbol{X} \end{pmatrix} = \bar{\boldsymbol{b}}, \qquad I \text{ 为 } m \times m \text{ 阶单位阵} \tag{4}$$

令　$\overline{A} = (I \ \vdots \ -A)$，$\overline{X} = (\bar{x}_1, \cdots, \bar{x}_m, \bar{x}_{m+1}, \cdots, \bar{x}_{m+n})^T = \begin{pmatrix} \boldsymbol{Y} \\ \boldsymbol{X} \end{pmatrix}$，则(LP)模型可写成：

$$\min Z = \sum_{j=1}^{m+n} c_j \cdot \bar{x}_j \tag{5}$$

$$\text{s.t.} \quad \overline{A} \cdot \overline{X} = \bar{b} \tag{6}$$

$$\bar{x}_j = y_j, \qquad j = 1,\cdots,m \tag{7}$$

$$c_j = 0, \qquad j = 1,\cdots,m \tag{8}$$

$$c_j = 1, \qquad j = m+1,\cdots,m+n \tag{9}$$

$$\bar{x}_j = x_{j-m}, \qquad j = m+1,\cdots,m+n \tag{10}$$

设 B 阵为此规划的一个可行基，记对应的基变量下标集为：$I_B = \{B_1, B_2, \cdots, B_m\}$，非基变量下标集为 $I_D = \{1,2,\cdots,m+n\} \backslash I_B$，非基变量对应的列构成的矩阵记为 $\mathbf{D}$，$\overline{\mathbf{X}}_B$ 为基变量构成的向量，$\overline{\mathbf{X}}_D$ 为非基变量构成的向量，则(6)式可写成：

$$\boldsymbol{B}\overline{\mathbf{X}}_B + \mathbf{D}\overline{\mathbf{X}}_D = \bar{\boldsymbol{b}}$$

$$\overline{\mathbf{X}}_B = \boldsymbol{B}^{-1}b - \boldsymbol{B}^{-1}\mathbf{D}\overline{\mathbf{X}}_D$$

且目标函数

$$z = \sum_{j=1}^{m+n} c_j \bar{x}_j = \sum_{j\in I_B} c_j \bar{x}_j + \sum_{j\in I_D} c_j \bar{x}_j$$

令 $C_B = (c_{B_1}, c_{B_2}, \cdots, c_{B_m})^T$，$C_D$ 为非基变量对应的 c_j 系数构成的向量，则有

$$Z = C_B^T(B^{-1}\bar{b} - B^{-1}\mathbf{D}\overline{\mathbf{X}}_D) + C_D^T\overline{\mathbf{X}}_D = C_B^T B^{-1}\bar{b} + (C_D^T - C_B^T B^{-1}\mathbf{D})\overline{\mathbf{X}}_D$$

则可行基 B 是(LP)规划最优基的充分条件是：

$$C_D^T - C_B^T B^{-1} D \geqslant 0$$

令 $\overline{A}_{\cdot j}$ 代表 $\overline{A}$ 的第 j 列，则有

$$\lambda_j = c_j - C_B^T B^{-1} \overline{A}_{\cdot j} \geqslant 0, \quad \forall\, j \in I_D \tag{11}$$

若已求得关于 N 列($N \geqslant m$)的切割矩阵 A 的(LP)规划的最优基 B，则只需判断下式：

$$\hat{\lambda} = \min(c_j - C_B^T B^{-1} \overline{A}_{\cdot j}) \geqslant 0, \quad \forall\, j \in I_D / \{1, \cdots, m + N\} \tag{12}$$

是否成立，就可知道对于剩下未知的切割列，B 是否仍为最优基。因为 $c_j = 1, \forall\, j \in I_D/\{1, \cdots, m + N\}$(由(9)式知)，所以，只需求解下面问题：

$$\max \quad C_B^T B^{-1} \overline{A}_{\cdot j}, \quad \forall\, j \in I_D / \{1, \cdots, m + N\} \tag{13}$$

即可知道(12)式是否成立，由 $\overline{A}$ 的构造形式知，$-\overline{A}_{\cdot j}, \forall\, j \in I_D/\{1, \cdots, m+N\}$ 对应于一种切割方式，所以，(13)式等价于下列问题：

$$\begin{aligned}
&\max \ -C_B^T B^{-1} \cdot S \\
&\text{s.t.} \quad \sum_{i=1}^{m} l_i \cdot s_i \leqslant L \\
&\qquad\quad 0 \leqslant s_i \leqslant b_i，\text{且 } s_i \text{ 为整数}, i = 1, \cdots, m \\
&\qquad\quad s = (s_1, s_2, \cdots, s_m)^T \quad \text{代表一种切割方式}
\end{aligned} \tag{14}$$

$C_B^T B^{-1}$ 即为通常所说的影子价格向量。(14)式所描述的问题是一个典型的背包问题，通过求解此背包问题，可得出一新的切割方式，笔者称此过程为背包列生成过程，若此切割方式使 $\hat{\lambda} \geqslant 0$，则对于未知的切割方式，基 B 仍为最优基，否则，以新的切割方式列向量作为换基列对单纯形表进行转轴运算，改进(LP)规划的目标函数。

2.2　初始可行基的构造

由上面上节可知，只需先构造一可行基，作为 A 阵的前 m 列，然后以此为基础，不断通过背包列生成过程就可得到新的切割列向量。因此，初始可行基的构造只需保证(LP)规划有解就行了，所以，我们用下面的方法构造 A 阵的前 m 列，每一列对应一种单一下料的切割方式，这是一个非奇异对角阵：

$$\begin{bmatrix}
\min\{b_1, \mathrm{int}(L/l_1)\} & 0 & \cdots & 0 \\
0 & \min\{b_2, \mathrm{int}(L/l_2)\} & \cdots & 0 \\
\vdots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & \min\{b_m, \mathrm{int}(L/l_m)\}
\end{bmatrix}$$

2.3 整数约束的处理

上面的几节，笔者是先不考虑 x_j 的整数约束，通过背包列生成过程来求解(LP) 规划，在得到(LP) 规划最优解后，我们可以对所有的生成列作整数规划(ILP)，或对(LP) 规划的最优解作圆整处理，从而得到(ILP) 规划的近似最优解。

2.4 算法步骤

step 1 按 2.2 节所述方法生成初始可行基，令 $A = (a_{ij})_{m\times m}$

$a_{ii} = \min\{b_i, \mathrm{int}(L/l_i)\}$， $i = 1,\cdots,m; a_{ij} = 0, \forall\ j \neq i$；令 $N = m$

step 2 用改进对偶单纯形法求解关于 A 阵的(LP) 规划。

step 3 求解背包问题(14) 式得新的切割方式向量，若此切割方式向量使(12) 式成立，转 step 5，否则，执行 step 4.

step 4 A 阵增加一列，把新的切割方式向量嵌入 A 阵作为第 $N+1$ 列，令 $N = N+1$，用改进单纯形法求解关于 A 阵的(LP) 规划，转 step 3.

step 5 用 Gomory 割平面法求解关于 A 阵的(ILP) 规划或对(LP) 规划最优解作圆整处理。

step 6 输出下料结果，结束。

3 计算实例

在此算法基础上，笔者用 C 语言开发了条材优化下料系统，并在全国几十家工厂推广应用，取得了很好的节材效果，比人工下料提高利用率约 5%，且算法收敛速度快，大部分算例在 286 以上机型十几秒钟即收敛。(算例见附录)

4 结 论

笔者提出了一种背包列生成算法，成功地解决了对巨大数量切割方式的优化选取问题，从工程应用上给出了求解下料模型近似最优解的实用算法。该算法结构简明、易于编程，且算法速度快节材效果显著，具有很大的推广价值。

参 考 文 献

1 Liu Fei, Ding Zhongqing, Cai Zhengjun. The problems in a kind of mathematical models for stock cutting ad their solution. In: Proc of 11 th ICPR. New York: Taylor & Francis, 1991. 643～646

2 Gilmore P C, Gomory R E. A lineary programming approach to the cutting stock problem-PART Ⅰ. Operations Research, 1963, 11: 863～888

3 Dyckhoff H. A New Linear Programming Approach to the Cutting Stock Problem. Opns Res, 1981, 29(6): 1 094 ～1 104

4 Sarker B R. An optimum solution for one-dimensional slitting problems: A Dynamic Programming Approach. J Opl Res Soc, 1988, 39(8): 749～755

5 闵仲求. 合理下料实用数学模型及其计算机软件研究. 系统工程理论与实践, 1992, 4: 36～41

6 袁忠良. 最优化下料方法与程序. 天津: 天津大学出版社, 1986. 132～136

7 袁忠良. 运筹学应用程序集. 北京: 清华大学出版社, 1989. 42～44

附　录

例 1　文献[7]公布的一个实际下料问题需原材料 46 根，本算法只需 45 根，余废料降低 76.9%. 数据如下：

原 材 料：长＝4000 mm

下料规格：

规格：	260	107	40	30
数量：	400	400	200	800

切割方式为：

1# 方式	260×15	40×1	30×2		余 0	需 18 根
2# 方式	30×133				余 10	需 2 根
3# 方式	40×1	30×132			余 0	需 3 根
4# 方式	260×2	107×30	40×6	30×1	余 0	需 13 根
5# 方式	260×13	40×14	30×2		余 0	需 6 根
6# 方式	260×10	107×10	40×3	30×7	余 0	需 1 根
7# 方式	260×7	40×11	30×58		余 0	需 1 根
8# 方式	260×9	40×3	30×12		余 1180	需 1 根

原材料总根数 45 根，材料利用率 99.33%

例 2　文献[6]公布的一个实际下料问题需原材料 33 根，本算法只需 32 根，余废料降低 80%. 数据如下：

原 材 料：长＝4000 mm

下料规格：

规格：	200	170	62	33
数量：	200	400	200	200

切割方式为：

1# 方式	200×20			余 0	需 10 根
2# 方式	170×23	33×2		余 24	需 6 根
3# 方式	62×64			余 32	需 1 根
4# 方式	170×6	62×47	33×2	余 0	需 1 根
5# 方式	170×22	62×1	33×6	余 0	需 11 根
6# 方式	170×11	62×12	33×42	余 0	需 1 根
7# 方式	62×56	33×16		余 0	需 1 根
8# 方式	170×3	62×10	33×62	余 824	需 1 根

原材料总根数 32 根，材料利用率 99.22%