数理解析研究所講究録
1114巻 1999年 233-241

段取り替え数最小化を考慮したカッティングストック問題の定式化と近似解法

京都大学大学院情報学研究科数理工学専攻
梅谷 俊治, 柳浦 睦憲, 茨木 俊秀
Shunji UMETANI, Mutsunori YAGIURA, Toshihide IBARAKI
{umetani, yagiura, ibaraki}@kuamp.kyoto-u.ac.jp

1 まえがき

カッティングストック問題は, 定型の素材 (ストック) から様々な大きさの製品を顧客の注文に応じて切出す問題であり, 切出しにかかる総費用の最小化を目的とする. 本問題は, 鉄鋼, 製紙, ガラス, 繊維業を初めとする多くの分野に応用を持つ古典的な組合せ最適化問題の1つであり, ストックの形状や製品の切出し方法, 製造コストのバランスにより, 様々な問題のバリエーションが存在する.

1つのストックから切出す製品の組合せはカッティングパターンと呼ばれる (以下, パターンと略す). カッティングストック問題の主要な製造コストには,

- ストック使用総量
- 製品の注文に対する製造の過不足
- パターン変更に伴う段取り替え回数

などが挙げられる. これまでは, ストックの材料費が製造コストの大半を占めていたため, 従来のカッティングストック問題では, 余剰ストックの最小化を目的とした定式化が主に取り上げられ, 線形計画法に基づく効率の良いアルゴリズムが開発されてきた [1][2]. しかし, 近年では, ストックの材料費が安くなる一方で, パターン変更に伴う段取り替え作業にかかる人件費や, 過製造により生じる在庫管理費が, 製造コストにおいて大きな割合を占めるようになり, これらの最小化が重要な課題として注目されている. そこで, 本研究では段取り替えの回数の最小化, すなわちパターン数の最小化を目的とした1次元カッティングストック問題の定式化と近似解法の提案を行う.

提案する近似解法は, メタ戦略に基づいており, また, パターン候補を適応的に生成する手法を組み込む事で計算効率の大幅な向上を図っている. 化学繊維産業における現実的な問題例に対する計算実験を行い, 短時間で実用的な解が得られることを確かめた.

2 問題の定式化

本研究で扱うカッティングストック問題では, ストック長 L, 製品数 m, 各製品長 $l_1, l_2, \ldots, l_m$, 及び各製品の注文数 $d_1, d_2, \ldots, d_m$ が与えられる. ストックから製品を切り出した後の余剰部分を切り残しと呼ぶ. 図1にカッティングストック問題の例を示す.

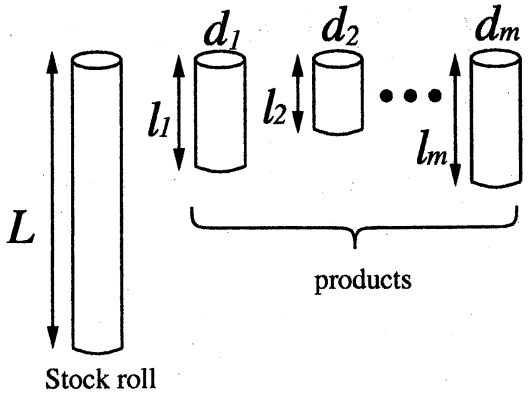

図 1: カッティングストック問題の例

パターンが切出す製品 i の数を $a_i \in \boldsymbol{Z}^+$ ($\boldsymbol{Z}^+$は非負整数の集合) とするとき,

$$\sum_{i=1}^{m} a_i l_i \leq L \tag{1}$$

を満たすパターンを利用可能パターンと呼ぶ. そのようなパターン全体の集合を S, パターン $j \in S$ をベクトル $(a_{1j}, a_{2j}, \ldots, a_{mj})$ で表す.

現実の問題では, 式 (1) の他に多くの制約条件が加わるため, 利用可能なパターン Sは制限される. 例えば, 切り出し機械のカッター数が制限されている場合には, 各パターンに含まれる製品数が制限される. また, ストックの切り残しコストの削減要求が厳しい場合には, 切り残し長が一定長以下のパターンのみが利用可能となる. 他にも, 生産上の特殊な制約が加わる場合が多くある.

使用パターンの集合をΠ, 各使用パターン j ($\in \Pi$) の適用回数を x_jとすると, 段取り替え数の最小化を目的とするカッティングストック問題は, 以下のように定式化できる.

$$\begin{aligned} (P) \quad & \min \quad |\Pi| \\ & \text{s.t.} \quad \sum_{i=1}^{m} \left(\sum_{j \in \Pi} a_{ij} x_j - d_i \right)^2 \leq D \\ & \Pi \subseteq S,\ x_j \in \boldsymbol{Z}^+ \end{aligned}$$

Dは注文に対する過不足の許容範囲を表す.

3 アルゴリズムの概略

問題 Pは制約条件が厳しいため, 目的関数である使用パターン数 $|\Pi|$ を直接最小化するアルゴリズムでは精度の高い解を求めることが困難となる. そこで, 提案するアルゴリズムでは問題 Pに対して, 使用パターン数 $|\Pi|$ をパラメータ Kに一時的に固定して, 問題 Pの制約条件からなる制約充足問題 $P(K)$ の解 $(\Pi, \boldsymbol{x})$ を求める.

$$(P(K)) \qquad f(\Pi, \boldsymbol{x}) \leq D$$

$$f(\Pi, \boldsymbol{x}) = \sum_{i=1}^{m} \left(\sum_{j \in \Pi} a_{ij} x_j - d_i \right)^2$$
$$|\Pi| = K$$
$$x_j \in \boldsymbol{Z}^+$$

問題 $P(K)$ の解が見付かればパラメータ Kを減らして再び問題 $P(K)$ を解く．そうでなければ，パラメータ Kを増やして再び問題 $P(K)$ を解く．パラメータ Kの調整は2分探索法等を用いて行い，Kが最小となる問題 $P(K)$ の解を探す．

問題 $P(K)$ の解 $(\Pi, \boldsymbol{x})$ は1パターンの入れ替えを近傍とする反復局所探索法を用いて求める．局所探索法において，2節で定義した利用可能パターン集合 $\boldsymbol{S}$をあらかじめ全て生成した上で入れ替え候補とすると，候補数が問題規模に対して指数オーダーで増加するため，効率の良い探索が期待できない．そこで，提案するアルゴリズムでは，探索解に応じて少数の新たなパターンを逐次生成して入れ替え候補とするアプローチを採る．また，各使用パターンの適用回数 $\boldsymbol{x}$ は使用パターンΠを決定した後に2次計画法を用いて求める．

図 2に提案するアルゴリズムの概略を示す．

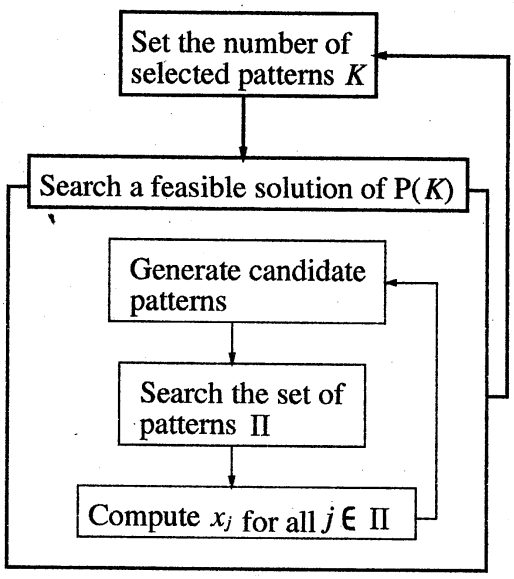

図 2: アルゴリズムの概略

4 使用パターンの適用回数

使用パターン集合Πが与えられたとき，各使用パターンについて問題 $P(K)$ の制約条件を満たす適用回数x_jか，そのようなx_jがない場合には，制約充足解に最も近いx_jを決定する必要がある．そこで，$f(\Pi, \boldsymbol{x})$ を最小化する以下の問題 $QP(\Pi)$ を定義する．

$$(QP(\Pi)) \quad \min \quad f(\Pi, \boldsymbol{x}) = \sum_{i=1}^{m} \left(\sum_{j \in \Pi} a_{ij} x_j - d_i \right)^2$$
$$\text{s.t.} \quad x_j \in \boldsymbol{Z}^+$$

問題$QP(\Pi)$は整数2次計画問題なので最適解を求めることは困難だが, $\boldsymbol{x}$の整数条件を緩和した問題$QP'(\Pi)$の実行可能領域は凸なので, 2次計画法を用いて高速に実数最適解$\boldsymbol{x}^*(\Pi)$を求めることができる. そこで, 緩和問題$QP'(\Pi)$の実数最適解$\boldsymbol{x}^*(\Pi)$を, 以下の方法により整数解に丸めて得られる解$\tilde{\boldsymbol{x}}(\Pi)$を問題$QP(\Pi)$の近似解とする.

整数解への丸めには, 局所探索法を用いる. まず, 実数最適解$\boldsymbol{x}^*(\Pi)$を四捨五入した解を初期解とし, 実数最適解$\boldsymbol{x}^*(\Pi)$を切り上げ, 切り下げして得られる解集合を探索する. 近傍は探索解$\boldsymbol{x}(\Pi)$から1変数$x_j(j \in \Pi)$を取り出し, $x_j = \lceil x_j^* \rceil$ ならば $x_j' \leftarrow x_j - 1$, $x_j = \lfloor x_j^* \rfloor$ ならば $x_j' \leftarrow x_j + 1$ として得られる解$\boldsymbol{x}'(\Pi)$の集合とする. 探索解は$f(\Pi, \boldsymbol{x})$により評価し, $f(\Pi, \boldsymbol{x}') < f(\Pi, \boldsymbol{x})$ ならば, $\boldsymbol{x}(\Pi) \leftarrow \boldsymbol{x}'(\Pi)$と新たな解に移る.

問題$QP'(\Pi)$の解法として, ここでは実装の比較的容易なガウス・ザイデル法 [3] を用いる. $\alpha_{pq} = 2\sum_{k=1}^{m} a_{kp}a_{kq}$, $\beta_p = 2\sum_{k=1}^{m} a_{kp}b_k$とおくと, $f(\Pi, \boldsymbol{x})$に対する勾配$\nabla f(\Pi, \boldsymbol{x})$の各要素$\frac{\partial f}{\partial x_p}$ $(p \in \Pi)$は,

$$\frac{\partial f}{\partial x_p} = \alpha_{p1}x_1 + \alpha_{p2}x_2 + \ldots + \alpha_{pn}x_n - \beta_p$$

と表せる. また, εは十分に小さい正定数とする.

$QP'(\Pi)$に対するガウス・ザイデル法

Step 0 全てのpに対して$x_p \leftarrow 0$, $\frac{\partial f}{\partial x_p} \leftarrow -\beta_p$とする.

Step 1 全てのpに対して以下の1,2いずれかの条件が成り立てば終了.

1. $|\frac{\partial f}{\partial x_p}| < \varepsilon$
2. $\frac{\partial f}{\partial x_p} > 0$ かつ $x_p = 0$

Step 2 Step1の条件を満たさないqを1つ選び, $\Delta x_q \leftarrow \frac{1}{\alpha_{qq}}\left(-\frac{\partial f}{\partial x_q}\right)$, $x_q \leftarrow \max(0, x_q + \Delta x_q)$ とする.

Step 3 全てのpに対して$\frac{\partial f}{\partial x_p} \leftarrow \frac{\partial f}{\partial x_p} + \alpha_{pq}\Delta x_q$とする. Step1に戻る.

5 カッティングパターンの生成

GilmoreとGomoryは列生成法と呼ばれるパターン生成法を提案しており [1][2], 余剰ストックの最小化を目的としたカッティングストック問題で成果をあげている. しかし, 列生成法はパターンの追加のみを想定したアプローチであり, パターンの入れ替えを行う本アルゴリズムへの適用は困難である. そこで, 本節では新たなパターン生成法を提案する.

探索解$(\Pi, \boldsymbol{x})$から, パターン$j \in \Pi$を除くと($\boldsymbol{x}$は変えないものとする), 幾つかの製品に不足が生じる. このときの製品iの残り需要を

$$d_i'(\Pi \setminus \{j\}, \boldsymbol{x}) = \max\left(d_i - \sum_{k \in \Pi \setminus \{j\}} a_{ik}x_k, 0\right)$$

と定義する. 以下では, $d_i'(\Pi \setminus \{j\}, \boldsymbol{x})$ を $d_i'(j)$ と略す. これら残り需要に対する過不足を最小化するパターンの生成を考える. 新たに生成するパターンを$\boldsymbol{a}' = (a_1', a_2', \ldots, a_m')$, その適用回数を

x'と記す. この時, 各製品の過不足を最小化するパターン $\boldsymbol{a}' \in S$とその適用回数 x'を求める問題 $P'(\Pi \setminus \{j\}, \boldsymbol{x})$ は, 以下の通りとなる.

$$
\begin{aligned}
(P'(\Pi \setminus \{j\}, \boldsymbol{x})) \quad \min \quad & \sum_{i=1}^{m} (a_i' x' - d_i'(j))^2 \\
\text{s.t.} \quad & \sum_{i=1}^{m} a_i' l_i \leq L \\
& x' \in \boldsymbol{Z}^+, a_i' \in \boldsymbol{Z}^+
\end{aligned}
$$

問題 $P'(\Pi, \boldsymbol{x}, j)$ は整数計画問題なので, 最適解を求めることは困難だが, $\boldsymbol{a}'$と x'を実数緩和した問題ならば, 各製品の過不足を最小化する切り残しのないパターンは

$$
a_i^* = \left(\frac{L}{\sum_{i=1}^{m} d_i'(j) l_i} \right) d_i'(j), \; x^* = \frac{\sum_{i=1}^{m} d_i'(j) l_i}{L}
$$

となる. そこで, 実数最適解 $\boldsymbol{a}^*$を整数に丸めて得られるパターンを, 新たなパターン $\boldsymbol{a}'$として生成する. ここで $x' \leftarrow x^*$に固定すると, $\boldsymbol{a}^*$の最適な丸め方を求める問題 P_rは以下の通りとなる.

$$
\begin{aligned}
(P_r) \quad \min \quad & \sum_{i=1}^{m} (a_i' - a_i^*)^2 \\
\text{s.t.} \quad & \sum_{i=1}^{m} a_i' l_i \leq L \\
& a_i' \in \{\lceil a_i^* \rceil, \lfloor a_i^* \rfloor\}
\end{aligned}
$$

問題 P_rは, 各変数 a_i'の取り得る値が2値なので, 以下の 0-1 ナップサック問題 P_r'と等価である.

$$
\begin{aligned}
(P_r') \quad \min \quad & \sum_{i=1}^{m} (1 - 2(a_i^* - \lfloor a_i^* \rfloor)) y_i \\
\text{s.t.} \quad & \sum_{i=1}^{m} y_i l_i \leq L - \sum_{i=1}^{m} \lfloor a_i^* \rfloor l_i \\
& y_i \in \{0, 1\}
\end{aligned}
$$

ここで, 問題 P_r'において $y_i = 0$ ならば $a_i' \leftarrow \lfloor a_i^* \rfloor$, $y_i = 1$ ならば $a_i' \leftarrow \lceil a_i^* \rceil$ とすれば, 問題 P_rの等価な解が得られる. 0-1 ナップサック問題の最適解を高速に求めるアルゴリズムは幾つか提案されているが, 問題 P_r'の最適解が必ずしも有用なパターンであるとは限らないので, 貪欲法を基本とした解法 [5] を用いて複数の近似解を求め, パターン候補とする. 以下に, 貪欲法と問題 P_r'に対する解法を示す.

貪欲法

Step0 製品 $1, 2, \ldots, m$ を$(1 - 2(a_i^* - \lfloor a_i^* \rfloor))/l_i$の小さい順に整列する.

その順序を$\sigma(1), \sigma(2), \ldots, \sigma(m)$ とする.

Step1 $k \leftarrow 1$ とする.

Step2 $\sum_{i=1}^{k} y_{\sigma(i)} l_{\sigma(i)} \leq L - \sum_{i=1}^{m} \lfloor a_{\sigma(i)}^* \rfloor l_{\sigma(i)}$ならば $y_{\sigma(k)} \leftarrow 1$ に固定して Step3 に進む.

Step3 $k \geq m$ ならば終了する, そうでなければ $k \leftarrow k+1$ として Step2 に戻る.

問題 P'_r に対する解法

Step0 貪欲法を用いて解 $(y_1^*, y_2^*, \ldots, y_m^*)$ を求めて, 出力する.

Step1 $i \leftarrow 1$ とする.

Step2 Step0 で求めた解が, $y_i^* = 0$ ならば, $y_i \leftarrow 1$ とする. そうでなければ, $y_i \leftarrow 0$ とする.

Step3 貪欲法を用いて $y_j (j \neq i)$ を求めて, 解 $(y_1, y_2, \ldots, y_m)$ を出力する.

Step4 $i \geq m$ ならば終了. そうでなければ, $i \leftarrow i+1$, として Step2 に戻る.

以上のアルゴリズムにより $m+1$ 個のパターン候補から成る集合 $S'(\Pi, j)(\subseteq S)$ が生成される.

ここで, 探索解 $(\Pi, \boldsymbol{x})$ と $j \in \Pi$ に対し, Π からパターン $j \in \Pi$ を除き, $j' \in S'(\Pi, j) \setminus \Pi$ を加えた時の傾向を数値実験により調べた.

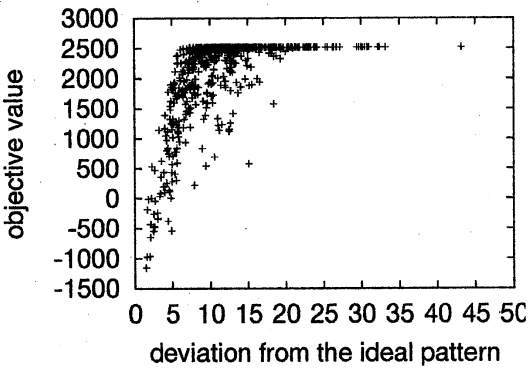

図 3: 全利用可能パターンに対する $f(\Pi, x)$ の変化量

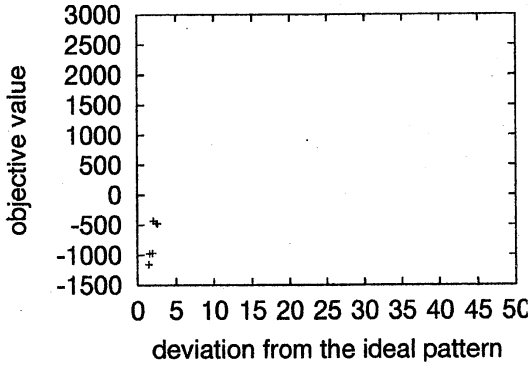

図 4: 生成パターンに対する $f(\Pi, x)$ の変化量

図 3,4の各点は 1 つのパターン候補に対応する. 縦軸はパターン入れ替え時の $f(\Pi, \boldsymbol{x})$ の変化量 $\Delta f = f(\Pi \cup \{j'\} \setminus \{j\}, \boldsymbol{x}') - f(\Pi, \boldsymbol{x})$ を, 横軸は各パターン候補 $\boldsymbol{a}_{j'}$ と問題 $P'(\Pi \setminus \{j\}, \boldsymbol{x})$ の実数最適解 $\boldsymbol{a}^*$ との乖離度 $\sum_{i=1}^{m}(a_{ij'} - a_i^*)^2$ を表す. 図 3より, 実際に改善解が得られるパターン候補の割合は非常に小さく, 全ての利用可能パターンを入れ替え候補としたのでは効率良い探索は期待できない事が分かる. また, 乖離度 $\sum_{i=1}^{m}(a_{ij'} - a_i^*)^2$ と目的関数の変化量との間に強い正の相関があることが観測できる. このことは, 問題 P_r に基づいてパターン候補を生成する方法の有効性を示唆している. また, 図 4より, 提案するパターン生成法が実際に改善の得られるパターン候補を生成している事が分かる.

6 局所探索法

使用パターン集合 Π の近傍 $NB(\Pi)$ を

$$NB(\Pi) = \{\Pi \cup \{j'\} \setminus \{j\} \mid j' \in S' \setminus \Pi, j \in \Pi\}$$

と定義する. 局所探索法の初期解 $(\Pi^{init}, \boldsymbol{x}^{init})$ は以下の方法により生成する. まず, $\Pi = \emptyset$, $\boldsymbol{x} = 0$ とする. 次に, $d_i'(\Pi \setminus \{j\}, \boldsymbol{x})$ の代わりに $d_i'(\Pi, \boldsymbol{x})$ を残り需要として 5節のパターン生成法を適用

する. 生成されたパターン候補を加えた場合の $\boldsymbol{x}(\Pi \cup \{j'\})$ を計算し, $f(\Pi \cup \{j'\}, \boldsymbol{x}(\Pi \cup \{j'\}))$ を評価する. パターン候補の中で最も $f(\Pi \cup \{j'\}, \boldsymbol{x}(\Pi \cup \{j'\}))$ の小さいパターンを選択し, Πに加える. 以上の手続きをパターン数が $|\Pi| = K$ となるまで繰り返す.

反復局所探索法のアルゴリズムを以下に示す. 各使用パターン $j \in \Pi$を初期解生成時に生成された順に$\sigma(1), \sigma(2), \ldots, \sigma(K)$ とする. $trials$ は改善なしのパターン入れ替えを行った回数を, $MAXTRIALS$は $trials$ の上限を表す.

反復局所探索法 ILS

Step 0 $trials \leftarrow 0$, $\Pi^* \leftarrow \Pi^{init}$とする.

Step1 近傍$NB(\Pi)$ を探索する.

- **Step1-1** $i \leftarrow 1$ とする.
- **Step1-2** 探索解 $(\Pi, \boldsymbol{x})$ からパターン$\sigma(i)$ を除き, パターン候補 S'を生成する.
- **Step1-3** 任意のパターン候補 $j' \in S'$を加えた$\Pi' \leftarrow \Pi \cup \{j'\}(\backslash\{\sigma(i)\})$ に対して $\boldsymbol{x}(\Pi')$ を計算し, $f(\Pi', \boldsymbol{x}(\Pi'))$ が最小となるパターン候補 j'を求める.
- **Step1-4** 解 $(\Pi', \boldsymbol{x}(\Pi'))$ が問題 $P(K)$ の制約条件を満たすならば, それを出力して終了. $f(\Pi', \boldsymbol{x}(\Pi')) < f(\Pi, \boldsymbol{x}(\Pi))$ ならば$\Pi \leftarrow \Pi'$, $i \leftarrow 0$ として Step1 に戻る.
- **Step1-5** $i \leftarrow i+1$ とする. $i \geq K$ならば Step2 に行く. そうでなければ Step1-1 に戻る.

Step 2 $f(\Pi, \boldsymbol{x}(\Pi)) < f(\Pi^*, \boldsymbol{x}(\Pi^*))$ ならば, $\Pi^* \leftarrow \Pi'$とする.

Step 3 $trials \geq MAXTRIALS$ならば終了する. そうでなければ, ランダムに$\Pi' \in NB(\Pi^*)$ を選び$\Pi \leftarrow \Pi'$, $trials \leftarrow trials + 1$ として, Step1 に戻る.

7 数値実験

本節では, 提案するアルゴリズムに対する数値実験の結果を記す. アルゴリズムはC言語で記述し, 数値実験は PentiumII(450MHz) プロセッサ搭載の PC/AT 互換機上で行った. 問題例は, 化学繊維産業における実例を用いた. 表 1に数値実験で用いた問題例の 1 つを示す.

表 1: カッティングストック問題の例

ストック長：2400		
番号	製品長	オーダ数
1	501	120
2	475	111
3	438	62
4	420	106
5	389	72
6	368	11
7	360	82
8	352	141
9	347	111
10	312	134

パターン数最小化を考慮したカッティングストック問題に対するアプローチとしては, Haessler らによる SHP(Sequential Heuristic Procedure) がある [4]. そこで, SHP についても提案するアルゴリズムと同様の数値実験を行い, 性能比較を行った.

ストック長 9000mm, 製品数 6～29 の例題に対して, 提案するアルゴリズム ILS と SHP を適用した結果を表 2に示す. 表 2より分かるように, ほとんど全ての例題において提案するアルゴリズムの解の精度が SHP のそれを上回っている. 特に, SHP では, パターン数が極端に多くなる例題や, 計算時間が非常に長くなる例題が幾つか見られるが, 提案するアルゴリズムでは, いずれの例題に対しても安定した性能を維持していることが分かる.

表 2: 計算結果 (ストック長 9000mm)

製品数	パターン数		計算時間 (秒)	
	ILS	SHP	ILS	SHP
6	2	8	0.02	0.09
7	3	3	0.02	0.02
8	3	4	0.01	0.05
9	3	4	0.08	0.04
10	4	5	0.27	0.63
11	4	4	0.17	0.86
13	4	8	0.17	1.32
13	3	4	0.05	0.62
14	4	5	0.18	0.12
15	5	6	1.20	0.74

製品数	パターン数		計算時間 (秒)	
	ILS	SHP	ILS	SHP
16	5	6	0.55	0.82
17	5	12	0.69	95.57
18	5	6	0.65	0.57
19	6	9	0.90	3.77
20	4	6	0.13	2.21
23	7	9	2.63	3.75
26	9	8	7.64	4.47
28	7	7	3.03	5.63
28	9	14	5.92	3021.66
29	7	10	2.23	679.9

次に, ストック長のみを 5000mm に変更した例題に対して, 提案するアルゴリズム ILS と SHP を適用した結果を表 3に示す. 以下の表 3では, 全例題において, 提案するアルゴリズム ILS に比べて SHP のパターン数が極端に多くなっている事が分かる. SHP では, 残り注文数の最も多い製

品をなるべく多く取るパターンが生成されるため, ストック長が短い場合には, 同一の製品のみから成るパターンが頻繁に生成され, パターン数がほぼ製品数程度となってしまう事が原因である.

以上の数値実験の結果より, 提案するアルゴリズムは従来のアプローチである SHP に比べ, より少ないパターン数の解を得ることができ, また, 多くの例題に対して, 解の精度と計算時間がより安定している事が確認できた.

表 3: 計算結果 (ストック長 5000mm)

製品数	パターン数		計算時間 (秒)	
	ILS	SHP	ILS	SHP
6	3	7	0.01	0.02
7	3	10	0.06	0.30
8	4	10	0.05	0.23
9	4	11	0.05	0.25
10	5	13	0.27	0.41
11	4	10	0.09	0.59
13	5	15	0.31	0.60
13	3	11	0.05	0.29
14	5	14	0.28	0.48
15	5	15	0.09	2.95

製品数	パターン数		計算時間 (秒)	
	ILS	SHP	ILS	SHP
16	6	10	0.77	0.16
17	6	18	0.43	4.32
18	7	20	1.22	2.02
19	8	23	1.02	5.19
20	6	8	0.39	0.32
23	10	20	2.27	1.87
26	11	27	9.72	33.19
28	9	10	3.10	0.80
28	10	20	5.01	5.44
29	9	10	4.97	0.56

8 おわりに

本研究では, 段取り替え数最小化を考慮したカッティングストック問題の定式化を行った. また, 探索解に基づいて適応的にカッティングパターンを生成する手法を提案し, これをメタ戦略に組み込むことで, 効率良い近似解法を提案した. 化学繊維産業の実例に対する数値実験の結果, 提案手法は, 少ない段取り替え数の解を短時間で求めることができ, 十分な実用性があることが確かめられた.

参考文献

[1] P. C. Gilmore and R. E. Gomory: A Linear Programming Approach to the Cutting-Stock Problem, Operations Research, Vol. 9, No. 6, pp. 849–859 (1961)

[2] P. C. Gilmore and R. E. Gomory: A Linear Programming Approach to the Cutting-Stock Problem – Part II, Operations Research, Vol. 11, No. 6, pp. 863–888 (1963)

[3] D. P. Bertsekas: Nonlinear Programming, Athna Scientific, (1995)

[4] R. W. Haessler: Controlling Cutting Pattern Changes in One-Dimentional Trim Problems, Operations Research, Vol. 23, No. 3, pp. 483–493 (1975)

[5] S. Sahni: Approximate Algorithms for the 0/1 Knapsack Problem, Jounal of the Association for Computing Machinery, Vol. 22, No. 1, pp. 115–124 (1975)