数理解析研究所講究録 1297 巻 2002 年 116-124

カッティングストック問題に対する線形計画法に基づく局所探索法の提案

京都大学·情報学研究科　梅谷 俊治　(Shunji Umetani)
柳浦 睦憲　(Mutsunori Yagiura)
茨木 俊秀　(Toshihide Ibaraki)
Graduate School of Informatics,
Kyoto University

Abstract

カッティングストック問題は, 様々な形状や大きさの製品を, 顧客の注文に応じて定型の母材から切出し, 母材の材料費や切出しにかかる工程費を最小化する計画を求める問題である. 1本の母材から切出す製品の組合せはカッティングパターン (以下, パターンと略す) と呼ばれる. 過去の素材産業では材料費が総費用の大半を占めていたため, 単に使用する母材の本数を最小化する定式化に基づく手法が多く提案されてきた. しかし, 近年, 材料費の低下に伴いパターン変更に伴う段取り替え作業回数の削減が重要視されている. そこで, 本研究では, あらかじめ与えられた段取り替え数 (パターン数) の元で, 使用母材の本数を最小化するカッティングストック問題に対して, 線形計画法に基づく局所探索法を提案する.

1　カッティングストック問題

入力として, 母材長 L, 製品種類数 m, 各製品の長さ $l_1, l_2, \ldots, l_m$, 及び注文数 $d_1, d_2, \ldots, d_m$ と, 使用可能なパターン数 n が与えられる. パターン p_j が製品 i を切出す数を a_{ij}, パターン p_j の適用回数を x_j とすると, 使用可能パターン数 n の元で使用母材の本数を最小化するカッティングストック問題は以下の通りに定式化できる.

$$\begin{aligned} (\mathrm{CSP}) \quad \min \quad & f(\Pi, X) = \sum_{p_j \in \Pi} x_j && (1) \\ \text{s.t.} \quad & \sum_{p_j \in \Pi} a_{ij} x_j \leq d_i, \text{ for } i \in M \\ & \Pi \subseteq S \\ & |\Pi| = n \\ & x_j \in \boldsymbol{Z}_+, \text{ for } p_j \in \Pi \\ & r_i \in \boldsymbol{Z}_+, \text{ for } i \in M. \end{aligned}$$

ここで, M は製品の集合 $\{1, 2, \ldots, m\}$, Π は使用パターンの集合 $\{p_1, p_2, \ldots, p_n\}$, X は使用パターン集合 Π の各パターンの適用回数 $\{x_1, x_2, \ldots, x_n\}$ を表す. S は以下の式 (2) を満たす任意のパターンから成る集合を表す.

$$\sum_{i \in M} a_{ij} l_i \leq L. \qquad (2)$$

カッティングストック問題は NP 困難のクラスに属する事が知られている.

2 提案解法

本研究では, カッティングストック問題に対して局所探索法に基づく解法を提案する. 局所探索法は, ある初期実行可能解から開始して, 近傍操作により得られる近傍解に暫定解より良い解があれば暫定解を入れ替える作業を繰り返し行う解法であり, NP 困難に代表される複雑な組合せ最適化に対する代表的な近似解法として多くの問題に適用されている. 局所探索法を組合せ最適化問題に適用する際には, 以下の点を考慮する必要がある.

- 初期実行可能解の生成
- 近傍解の生成
- 近傍解の評価

使用パターン数に制限のあるカッティングストック問題では, 初期実行可能解を求める問題は NP 完全のクラスに属するビンパッキング問題と等価である. 本研究では 2.1節で, ビンパッキング問題に対する近似解法の 1 つである First-Fit 法を適用して初期実行可能解を求める. 次に, 近傍解 $\Pi' \in N(\Pi)$ は, 使用パターン集合Π内のあるパターン $p_j \in \Pi$を新たなパターン $p'_j \in S$に入替えて得られる. しかし, 式 (2) に示される実行可能なパターンの数は製品数 m の指数オーダーとなるため, これら全てを暫定使用パターン集合に対する入替えパターン候補として評価することは困難である. 本研究では 2.2節で, 線形計画問題における感度分析を基に, 改善解が得られる見込みのあるパターンのみを生成する手法を提案し, 暫定解に対して $O(n^2)$ 個の入替えパターン候補を生成する. 各パターンの適用回数 Xは, 使用パターン集合Πが与えられた際に以下の式 (3) の整数計画問題を解く事で得られる.

$$
\begin{aligned}
(\mathrm{IP}(\Pi)) \quad \min \quad & f(X) = \sum_{j=1}^{n} x_j \\
\text{s.t.} \quad & \sum_{j=1}^{n} a_{ij} x_j - r_i = d_i, \text{ for } i \in M \\
& x_j \in \boldsymbol{Z}_+, \text{ for } j \in N \\
& r_i \in \boldsymbol{Z}_+, \text{ for } i \in M.
\end{aligned} \tag{3}
$$

提案解法では, 式 (3) の各変数 x_jの整数制約 ($x_j \in \boldsymbol{Z}_+$) を非負実数制約 ($x_j \geq 0$) に緩和して得られる線形計画問題 LP(Π) を解き, 得られた実数最適解$\overline{X}$ を丸めて得られる整数近似解$\hat{X}$を各パターンの適用回数とする. しかし, 依然として非常に多くの線形計画問題を繰り返し解く必要があるため, 本研究では 2.3節で線形計画法の高速化を行う. 最後に局所探索法の概観を 2.4節でまとめる.

2.1 初期解生成

本節では, 初期実行可能解を求める発見的解法について説明する. 使用パターン数が n 種類に制限されたカッティングストック問題において, 初期実行可能解を求める問題は, 各製品 $i \in M$が使用パターン集合$\Pi = \{p_1, p_2, \ldots, p_n\}$ に少なくとも 1 度現れる割当てを求める問題として定式化できる. これは, 以下のビンパッキング問題と等価である.

ビンパッキング問題

入力: 製品集合 $M=\{1,2,\ldots,m\}$ と各製品長 l_i, 使用パターン数 n, 母材長 L.

出力: 製品集合 Mの分割 $M_1,M_2,\ldots,M_n$, 各部分集合 $M_j \subseteq M$は$\sum_{i\in M_j} l_i \leq L$ を満す.

本研究では, ビンパッキング問題に対する発見的解法である First-Fit 法を基本とした発見的解法 INIT を提案する. INIT を実行する前に全使用パターン $p_j \in \Pi$を空に初期化する. 初めのパターン p_1では, 全製品 $i \in M$を注文数 d_iの降順に整列した後に順番に割当てる. 以降の各パターン $p_j(j \geq 2)$ では, 全製品 $i \in M$をこれまでに現れた回数$\lambda_i = \sum_{q=1}^{j-1} a_{iq}$の昇順で整列した後に順番に割当てる. INIT は各パターン $p_j \in \Pi$に各製品 $i \in M$を 1 個ずつしか割当てないので, 母材長に対して各製品長が小さい場合には各パターンに大きな切残しができる. 以下にアルゴリズム INIT を示す. ここで L_j^{res}はパターン p_jの切残し長を表す.

アルゴリズム INIT

入力: 各製品の長さ l_iと注文数 d_i, 使用パターン数 n, 母材長 L.

出力: 使用パターン集合$\Pi = \{p_1,p_2,\ldots,p_n\}$.

Step 1: 全パターン $p_j \in \Pi$を空にする $(\forall i\ a_{ij} := 0)$. $j := 1$, $L^{res} := L$ とする.

Step 2: $j = 1$ ならば全製品 $i \in M$を注文数 d_iの降順に整列する. そうでなければ, 全製品 $i \in M$を出現回数$\lambda_i = \sum_{q=1}^{j-1} a_{iq}$の昇順に整列する. $\sigma(k)$ を整列後の k番目の製品とする. $k := 1$ とする.

Step 3: $l_{\sigma(k)} \leq L_j^{res}$ならば $a_{\sigma(k)j} := 1$, $L_j^{res} := L_j^{res} - l_{\sigma(k)}$とする. $k < m$ ならば $k := k+1$ として Step 3 に戻る.

Step 4: $j = n$ ならば使用パターン集合Πを出力して終了. そうでなければ, $j := j+1$ として Step 2 に戻る.

2.2 パターン生成

提案する局所探索法では, 使用パターン集合Πの探索に, 以下に示す 1 パターン入替え近傍 $N(\Pi)$ を用いる.

$$N(\Pi) = \{\Pi \cup \{p(i',j')\} \setminus \{p_j\} \mid p_{j'} \in \Pi, i' \in M'(\Pi)\}, \tag{4}$$

$p(i',j')$ は後で説明するアルゴリズム PATGEN で生成されるパターンを表す. また, $M'(\Pi)$ は製品集合 Mの部分集合で, 以下の式 (5) で定義される.

$$M'(\Pi) = \{i \mid \overline{y}_i > 0, i \in M\}. \tag{5}$$

ここで, $\overline{Y} = \{\overline{y}_1,\overline{y}_2,\ldots,\overline{y}_m\}$ は以下に示す線形計画問題 LP(Π) の双対問題 DLP(Π) の最適解である.

$$\begin{aligned} (\mathrm{DLP}(\Pi)) \quad \max \quad & \sum_{i=1}^{m} d_i y_i \\ \text{s.t.} \quad & \sum_{i=1}^{m} a_{ij} y_i \leq 1, \text{ for } p_j \in \Pi \\ & y_i \geq 0, \text{ for } i \in M. \end{aligned} \tag{6}$$

パターン生成アルゴリズム PATGEN は暫定解 (Π, X) 及び (i',j') を入力とし, パターン $p_{j'} \in \Pi$ を変更して得られるパターン候補 $p(i',j')$ を出力する. まず, パターン $p_{j'}$内の製品 i'を 1 増やし,

式(2)を満たすならば終了. そうでなければi'以外の製品を$\overline{y}_i/l_i$の値で整列し, $\overline{y}_i/l_i$の小さい製品から式(2)を満すまで順に1ずつ減らす. 以下の式(7)を満たすならば終了.

$$L - \sum_{i \in M} a_{ij} l_i < \min_{i \in M} l_i \tag{7}$$

そうでなければ, $\overline{y}_i/l_i$の大きい順に式(2)の制約のもとで製品iを加える.

アルゴリズム PATGEN

入力: 製品$i' \in M$, パターン$p_{j'} = (a_{1j'}, a_{2j'}, \ldots, a_{mj'})$, 各製品の長さ$l_i$と注文数$d_i$, 使用パターン数$n$, 母材長$L$. 双対問題DLP(Π)の最適解$\overline{Y} = \{\overline{y}_1, \overline{y}_2, \ldots, \overline{y}_m\}$, 製品集合$M'(\Pi) \subseteq M$.

出力: パターン$p(i', j') = (a'_{1j'}, a'_{2j'}, \ldots, a'_{mj'})$.

Step 1: 全製品$i \in M$を$\overline{y}_i/l_i$の降順に整列する. $\sigma(k)$を整列後のk番目の製品とする. $k := 1$とする.

Step 2: $a'_{i'j'} := a_{i'j'} + 1$, $a'_{ij'} := a_{ij'} (\forall i \neq i')$とする. $L^{res} := L - \sum_{i \in M} a'_{ij'}$とする.

Step 3: $L^{res} \geq 0$ならば$k := m$としてStep 5へ行く. そうでない場合$k > m$ならば終了.

Step 4: $\sigma(k) \neq i'$かつ$a'_{\sigma(k)j'} > 0$ならば, $a'_{\sigma(k)j'} := a'_{\sigma(k)j'} - 1$, $L^{res} := L^{res} + l_{\sigma(k)}$とする. $k := k + 1$としてStep 3に戻る.

Step 5: $l_{\sigma(k)} \leq L^{res}$ならば$a'_{\sigma(k)j'} := a'_{\sigma(k)j'} + 1$, $L^{res} := L^{res} - l_{\sigma(k)}$. $k > 1$ならば$k := k - 1$としてStep 5に戻る, そうでなければパターン$p(i', j')$を出力して終了.

2.3 線形計画問題

提案する局所探索法では, 近傍解$\Pi' \in N(\Pi)$を評価するたびに線形計画問題LP(Π')を解く必要があるため, 単体法をそのまま適用したのでは, LP(Π')を解くたびに単体表を作り直す必要があり計算効率が悪い. そこで, 本研究では暫定解Πにおける各パターンの適用回数Xを計算する際に得られた単体表Tから, 近傍解Π'に対する単体表T'を生成し, これに対して線形計画法の一種である十文字法[4]を適用して, 近傍解Π'の各パターンの適用回数X'を計算する.

暫定解Πにおける単体表Tの基底変数に対する列集合を$\boldsymbol{B}$, コスト係数$\boldsymbol{c}_B$, 非基底変数に対する列集合を$\boldsymbol{N}$, コスト係数$\boldsymbol{c}_N$とする. 新たな単体表T'は, 2.2節で生成したパターン$p(i', j')$に$\boldsymbol{B}^{-1}$を掛けて得られる列$\tilde{p}(i', j') = \boldsymbol{B}^{-1} p(i', j')$を$\boldsymbol{N}$に加えた後, これとパターン$p_{j'}$に対応する列$\tilde{p}_{j'}$を入替えるピボット操作を適用することで得られる. この時, パターン$p_{j'}$に対応する相対コスト$\tilde{c}'_{j'}$は$\tilde{c}'_{j'} = \tilde{c}_{j'} - \sum_{i \in M} \overline{y}_i (a'_{ij'} - a_{ij'})$と計算できる. 得られた単体表$T'$は必ずしも実行可能ではないが, 任意の単体表から最適解に収束する性質を持つ十文字法[4]を適用する事で, 近傍解Π'に対する単体表T'が得られる.

アルゴリズム SOLVE_LP

入力: LP(Π)の最適解$\overline{X} = \{\overline{x}_1, \overline{x}_2, \ldots, \overline{x}_n\}$, 単体表$T = (\boldsymbol{B}, \boldsymbol{N}, \tilde{\boldsymbol{c}}_B, \tilde{\boldsymbol{c}}_N)$. パターン$p(i', j') = (a'_{1j'}, a'_{2j'}, \ldots, a'_{mj'})$.

出力: LP(Π')の最適解$\overline{X}' = \{\overline{x}'_1, \overline{x}'_2, \ldots, \overline{x}'_n\}$, 単体表$T = (\boldsymbol{B}', \boldsymbol{N}', \tilde{\boldsymbol{c}}'_B, \tilde{\boldsymbol{c}}'_N)$.

Step 1: 列$\tilde{p}(i',j') := \boldsymbol{B}$, $\overline{x}'_{j'} := 0$, $\tilde{c}'_{j'} := \tilde{c}_{j'} - \sum_{i\in M} \overline{y}_i(a'_{ij'} - a_{ij'})$ を単体表Tの非基底列集合$\boldsymbol{N}$に加える.

Step 2: ピボット操作を施し列$\tilde{p}(i',j')$と$\tilde{p}_{j'}$を入れ替えた後, 十文字法を適用して新たな最適解$\overline{X}'$およびその単体表T'を得て終了.

2.4 局所探索法

本節では2.1–2.3節のINIT, PATGEN, SOLVE_LP を用いた局所探索法の概要をまとめる. (Π, X)を暫定解とする. 提案する局所探索法では, 暫定解Πの近傍$N(\Pi)$に改善解Π'が見つかった場合には, 即座に暫定解Πを改善解Π'で置き換える. また, 解を評価する指標として目的関数$f(\Pi, X)$以外に, 余剰製品数の合計$over(\Pi, X) = \max\{0, \sum_{i\in M} a_{ij}x_j - d_i\}$を用いる. つまり, $f(\Pi', X') < f(\Pi, X)$, もしくは$f(\Pi', X') = f(\Pi, X)$かつ$over(\Pi', X') < over(\Pi, X)$を満す近傍解$(\Pi', X')$を改善解と見なす.

アルゴリズム LS

入力: 各製品の長さl_iと注文数d_i, 使用パターン数n, 母材長L.

出力: 使用パターン集合$\Pi = \{p_1, p_2, \ldots, p_n\}$, 各パターンの適用回数$X = \{x_1, x_2, \ldots, x_n\}$.

Step 1: アルゴリズム INIT を適用して初期使用パターン集合$\Pi = \{p_1, p_2, \ldots, p_n\}$を得る. SOLVE_LP を適用して各パターンの適用回数$X = \{x_1, x_2, \ldots, x_n\}$を得る. 実行不可能解が出力されたら終了. $i^* := 1$, $j^* := 1$, $i' := i^*$, $j' := j^*$とする.

Step 2: $i' \in M'(\Pi)$ ならば, アルゴリズム PATGEN を適用してパターン$p(i', j')$を生成し, $\Pi' = \Pi \cup \{p(i', j')\} \setminus \{p_{j'}\}$とする. そうでなければStep 5 に行く.

Step 3: 使用パターン集合Π'に対してアルゴリズム SOLVE_LP を適用して, 各パターンの適用回数$X = \{x_1, x_2, \ldots, x_n\}$を求める. アルゴリズム SOLVE_LP が実行不可能解を出力した場合はStep 5 へ行く.

Step 4: $f(\Pi', X') < f(\Pi, X)$, もしくは$f(\Pi', X') = f(\Pi, X)$かつ$over(\Pi', X') < over(\Pi, X)$ならば, $(\Pi, X) := (\Pi', X')$, $i^* := i^* + 1 \pmod m$, $j^* := j^* + 1 \pmod n$, $i' := i^*$, $j' := j^*$としてStep 2 に戻る.

Step 5: $i' := i' + 1 \pmod m$とする. $i' = i^*$(全てのi'が走査済み) ならば, $j' := j' + 1 \pmod n$とする. $j' = j^*$(全てのi', j'が走査済み) ならば, 暫定解(Π, X)を出力して終了, そうでなければStep 2 へ戻る.

3 数値実験

例題生成アルゴリズム CUTGEN[2] により様々な問題例を作成した. CUTGEN は入力パラメータに$L, \nu_1, \nu_2, \bar{d}$を持ち, Lは母材長, $[\nu_1, \nu_2]$は母材に対する各製品長の割合, $\bar{d}$は各製品の注文数の平均を表す. 本実験では, 入力パラメータを組合せて18 クラスの問題例(各クラス100 題) を作成した. ここで, $L = 1000$, $m = 10, 20, 40$, $\bar{d} = 10, 100$, クラス 1–6 では$(\nu_1, \nu_2) = (0.01, 0.2)$, クラス 7–12 では$(\nu_1, \nu_2) = (0.01, 0.8)$, クラス 13–18 では$(\nu_1, \nu_2) = (0.2, 0.8)$と設定した.

まず, 2.3節で提案した十文字法を用いた手法の評価を行った. 局所探索法1 回の実行の間に, 線形計画問題 LP(Π) を解くのに要したピボット操作の平均回数を, 改訂単体法と提案する十文字法

を用いた方法で比較を行った (表 1). 表 1に示す様に, 改訂単体法は製品数にほぼ比例したピボット回数を要しているのに比較して, 提案手法ではほとんどの問題例において 1–2 回のピボット操作で最適解に収束している.

表 1: LP(II) を解くのに要した平均ピボット回数

class	m	$\bar{d}$	改訂単体法	提案解法
7	10	10	11.99	1.27
8	10	100	14.14	0.88
9	20	10	19.46	0.88
10	20	100	31.48	1.19
11	40	10	52.16	0.84
12	40	100	73.23	1.64

次に, 提案解法 LS とパターン数最小化を考慮した発見的解法である SHP[3], KOMBI[1] との比較を行った. SHP は使用パターンを逐次生成するアルゴリズムである. SHP はパターン候補を列挙した後に, 残り需要をなるべく満しかつ適用回数が大きくなるパターンを発見的に選択して割当てる. KOMBI は使用パターン数制限の無いカッティングストック問題に対する近似解を初期解として, 複数の使用パターンを繰り返し併合する事で使用パターン数を最小化する. SHP と LS は AT 互換機 (Pentium III 1GHz, 1GB Memory) 上で実行した. KOMBI の結果については文献 [1] から引用した. 1節で述べたように, LS において使用パターン数 n は入力パラメータとして与えられる. 本実験では, 使用パターン数 $n = \beta, \beta - 1, \beta - 2$ について LS を実行した. βは以下の式で与えられる.

$$\beta = \min \left\{ n_{shp}, n_{shp} - \lceil \bar{n}_{shp} - \bar{n}_{kombi} \rceil \right\}, \tag{8}$$

ここで n_{shp}, n_{kombi}は各例題に対して SHP, KOMBI を適用して得られる使用パターン数を, $\bar{n}_{shp}$, $\bar{n}_{kombi}$はそれらの 100 例題での平均値を表す.

表 2に SHP,KOMBI,LS の計算実験結果を示す. 表中の (95/100) 等は LS を 100 例題に実行した際に 95 例題において実行可能解が得られた事を表す. 表 2より, KOMBI を適用して得られる使用母材の本数は, SHP,LS のそれより少ない事が確かめられる. これは, KOMBI が使用パターン数制限の無いカッティングストック問題の近似解を初期解としているためである. 一方で, LS は入力パラメータである使用パターン数 n を変える事により, 様々な使用パターン数において近似解を得ることができる利点がある.

表 2: アルゴリズム SHP, KOMBI, LS の計算実験結果

class	m	$\bar{d}$	SHP		KOMBI		LS ($\bar{f}$)		
			$\bar{n}$	$\bar{f}$	$\bar{n}$	$\bar{f}$	$n=\beta$	$n=\beta-1$	$n=\beta-2$
1	10	10	4.25	11.62	3.40	11.49	12.54	14.72 (97/100)	17.74 (53/100)
2	10	100	6.33	111.81	7.81	110.25	113.09	113.87	116.25
3	20	10	5.89	22.37	5.89	22.13	23.93	24.53	26.11
4	20	100	8.98	218.94	14.26	215.93	220.34	221.29	223.57
5	40	10	9.03	43.60	10.75	42.96	48.03	50.09	52.18
6	40	100	13.45	430.79	25.44	424.71	436.06	438.18	440.60
7	10	10	10.33	52.19	7.90	50.21	51.32 (94/100)	50.57 (82/100)	51.72 (60/100)
8	10	100	11.46	520.36	9.96	499.52	507.94	511.57	501.76 (93/100)
9	20	10	19.22	97.96	15.03	93.67	96.88	98.51 (99/100)	99.72 (98/100)
10	20	100	21.74	973.63	19.28	932.32	946.93	946.48	952.55
11	40	10	36.29	187.37	28.74	176.97	185.92	187.24	189.64 (99/100)
12	40	100	40.53	1865.41	37.31	1766.20	1782.34	1784.14	1788.58
13	10	10	10.55	65.41	8.97	63.27	64.24 (98/100)	63.43 (87/100)	60.80 (49/100)
14	10	100	10.95	654.80	10.32	632.12	637.22 (99/100)	637.29 (97/100)	623.87 (90/100)
15	20	10	20.30	124.36	16.88	119.93	121.74 (97/100)	122.23 (95/100)	121.87 (91/100)
16	20	100	21.09	1240.39	19.91	1191.80	1207.24	1209.46	1198.64 (97/100)
17	40	10	38.52	238.06	31.46	224.68	236.15	236.69 (99/100)	238.44 (99/100)
18	40	100	40.38	2366.95	38.28	2242.40	2265.84	2268.52	2279.58

次に, 表 2において各解法の計算に要した時間を表 3に示す. LS は $n=\beta$の場合の計算時間を示す. 使用計算機の速度を考慮すると SHP, KOMBI の方が LS よりも短時間で解を得ている. しかし, LS でも 300 秒以内に解を得ることが可能であり実用上十分な性能だと考えられる.

表 3: アルゴリズム SHP, KOMBI, LS の計算時間

class	m	$\bar{d}$	SHP	KOMBI	LS
1	10	10	0.04	0.14	0.11
2	10	100	0.08	1.14	0.50
3	20	10	1.56	1.74	1.75
4	20	100	1.57	16.00	11.38
5	40	10	631.74	38.03	20.89
6	40	100	107.11	379.17	279.33
7	10	10	0.00	0.07	0.08
8	10	100	0.00	0.20	0.13
9	20	10	0.01	1.34	1.00
10	20	100	0.02	3.25	1.51
11	40	10	0.09	36.27	10.49
12	40	100	0.14	76.31	18.08
13	10	10	0.00	0.08	0.07
14	10	100	0.00	0.13	0.11
15	20	10	0.01	1.81	1.22
16	20	100	0.01	2.60	1.69
17	40	10	0.06	50.93	14.90
18	40	100	0.10	70.94	23.09

4 まとめ

過去の素材産業では材料費が総費用の大半を占めていたため, 使用母材の本数を最小化する定式化に基づく手法が多く提案されてきた. しかし, 近年では材料費の定価に伴い, 段取り替えによって生じる費用の削減が重要視されている. そこで本研究では, あらかじめ与えられた段取り替え数 (パターン数) の元で, 使用母材の本数を最小化するカッティングストック問題に対して, 1 パターン入れ替え近傍に基づく局所探索法を提案した. カッティングストック問題では, 近傍解の構成要素である実行可能なパターンが莫大な数に及ぶ. そこで, 提案解法では子問題の線形計画問題から得られる相対コストの情報を用いて, 改善解を構成できるパターン候補のみを用いた近傍を定義した. また, 近傍解ごとに子問題の線形計画問題を繰り返し解く必要があるため, 本研究では, 暫定解に対する単体表を利用して線形計画問題を高速に解く方法を提案した. 最後に数値実験を行い, 他の発見的解法には若干劣るものの様々な使用パターン数に対して良好な精度を持つ解が得られる事が確かめられた.

参考文献

[1] H. Foerster and G. Wäscher, Pattern reduction in one-dimensional cutting stock problems, *International Journal of Production Research*, vol.38, pp.1657–1676, 2000.

[2] T. Gau and G. Wäscher, CUTGEN1: a problem generator for the standard one-dimensional cutting stock problem, *European Journal of Operational Research*, vol.84, pp.572–579, 1995.

[3] R.E. Haessler, Controlling cutting pattern changes in one-dimensional trim problems, *Operations Research*, vol.23, pp.483–493, 1975.

[4] S. Zionts, The criss-cross method for solving linear programming problems, *Management Science*, vol.15, pp.426–445, 1969.