カッティングストック問題におけるパターン生成法について

02004704　京都大学　*梅谷 俊治　UMETANI Shunji
01704164　京都大学　柳浦 睦憲　YAGIURA Mutsunori
01001374　京都大学　茨木 俊秀　IBARAKI Toshihide

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り出す問題であり, 切出しにかかる総費用の最小化を目的とする. 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 (以下, パターンと略す) と呼ばれる. 従来のカッティングストック問題では, 余剰素材の最小化を目的とした定式化が中心であり, 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, パターン変更に伴う段取り替え作業の削減が, 製造効率の向上に関わる重要な課題として取り上げられている. そこで, 本研究では段取り替え数の最小化, 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新たな定式化とヒューリスティックに基づく近似解法を提案する.

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を $d_1, d_2, \ldots, d_m$ が与えられる. あるパターンで切出す製品 i の数を $a_i \in \mathbb{Z}^+$ ($\mathbb{Z}^+$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$\mathcal{S}$, パターン $j \in \mathcal{S}$をベクトル $(a_{1j}, a_{2j}, \ldots, a_{mj})$ で表す. 使用パターンの集合Πと各使用パターン j ($\in \Pi$) の適用回数 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, 以下の様に定式化できる.

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

Dは注文に対する過不足の許容範囲を表す.

3　アルゴリズムの概要

本手法では, 問題 Pに対して, 目的関数である使用パターン数 $|\Pi|$ 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以下の制約充足問題 $P(K)$ の解 (Π, x) を1パターン入れ替えを近傍とする反復局所探索法を用いて求める.

$$\begin{aligned} (P(K)) \quad & f(\Pi, x) \leq D \\ & f(\Pi, x) = \sum_{i=1}^{m} \left(\sum_{j \in \Pi} a_{ij} x_j - d_i \right)^2 \\ & x_j \in \mathbb{Z}^+ \end{aligned}$$

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.

4　カッティングパターンの生成

カッティングストック問題では, 利用可能なパターンの数は, 問題の規模に対して指数的に増加するため, これらのパターン全てを入れ替え候補としたのでは効率の良い探索は望めない. そこで, 暫定解における残り需要の情報を基に, 入れ替えパターン候補を逐次生成して, 近傍探索の高速化を実現している.

探索解 (Π, x) から, パターン $j \in \Pi$を除くと (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

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

と定義する. 以下では, $d_i'(\Pi \setminus \{j\}, x)$ 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 $a' = (a_1', a_2', \ldots, a_m')$, 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 $a' \in \mathcal{S}$ とその適用回数 x'を求める問題 $P'(\Pi \setminus \{j\}, x)$ は, 以下の通りとなる.

$$(P'(\Pi \setminus \{j\}, x)) \quad \min \quad \sum_{i=1}^{m} (a_i' x' - d_i'(j))^2$$

© 日本オペレーションズ・リサーチ学会. 無断複写・複製・転載を禁ず.

$$\text{s.t.} \quad \sum_{i=1}^{m} a'_i l_i \leq L$$
$$x' \in Z^+, a'_i \in Z^+$$

問題 $P'(\Pi \setminus \{j\}, \boldsymbol{x})$ 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.

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

問題 P_rは, 各変数 a'_iの取り得る値が2値なので, 0-1ナップサック問題に定式化できる. 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. しかし, 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 [3] を用いて複数の近似解を求め, パターン候補とする. 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$S'(\subseteq S)$が生成される.

5 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, $f(\Pi, \boldsymbol{x})$ 最小化する整数 2 次計画問題 $QP(\Pi)$ の近似解 x_jを求める. 問題 $QP(\Pi)$ の実数緩和問題 $QP'(\Pi)$ の最適解 x_j^*を 2 次計画法により求める. x_j^*を整数に丸めた値を各使用パターンの適用回数 x_jとする.

6 局所探索法

使用パターン集合Πの近傍 $NB(\Pi)$ を

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

と定義する. 局所探索法の初期解 (Π^{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 4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 j'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.

<u>反復局所探索法 ILS</u>

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 $\Pi' \leftarrow \Pi \cup \{j'\}(\setminus \{\sigma(i)\})$ 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 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, ヒューリスティックに基づく新たなカッティングパターン生成を提案し, メタ戦略と組み合わせて効率良い近似解法の提案を行った. 数値実験の詳細および考察については, 当日発表する予定である.

参考文献

[1] P. C. Gilmore, R. E. Gmomory, "A linear programming approach to the cutting-stock problems," *Operations Research*, Vol. 9, pp. 848-859, 1961.

[2] P. C. Gilmore, R. E. Gmomory, "A linear programming approach to the cutting-stock problems PART II," *Operations Research*, Vol. 11, pp. 863-888, 1963.

[3] S. Sahni, "Approximate Algorithms for the 0/1 Knapsack Problem," *Jounal of ACM*, Vol. 22, No. 1, pp. 115–124, 1975.

© 日本オペレーションズ・リサーチ学会. 無断複写・複製・転載を禁ず.