2001年度日本オペレーションズ・リサーチ学会
秋季研究発表会

配置コストをもつ長方形詰込み問題に対する局所探索法について

02005174　京都大学　*今堀 慎治　IMAHORI Shinji
01704164　京都大学　柳浦 睦憲　YAGIURA Mutsunori
01001374　京都大学　茨木 俊秀　IBARAKI Toshihide

1　はじめに

長方形詰込み問題とは, 様々な大きさの長方形を, 二次元平面上に互いに重ならないように配置する問題であり, 位置に対するコスト関数によって計算の複雑さは異なるが, 多くの場合NP困難であることが知られている. 本研究では, 一般的な配置コストをもつ長方形詰込み問題に対し, 局所探索法に基づくアルゴリズムを提案する. 配置コスト関数が一般的なので, これらをうまく設定することにより, 様々な配置問題やスケジューリング問題をこの形に定式化することができ, 非常に汎用性の高い問題となっている. 本研究では, 順列対表現 [1] を用いて解を表現し, 順列対から配置を求める2つのアルゴリズム, および配置から順列対を求めるアルゴリズムを提案する. 順列対から配置を求めるアルゴリズムは動的計画法に基づいており, 文献 [3] が対象とする特別な配置コストに対する計算時間が, 文献 [3] のアルゴリズムとオーダー的に等価である上, より一般的な配置コストを扱うことができる点に特徴がある. さらに, 対象問題に応じて一般化されたクリティカルパスを定義し, これを利用して局所探索法における近傍の探索を効率化する工夫を行っている. 計算実験を行い, 他の手法との性能比較を行った.

2　問題の定義

長方形集合 $I = \{1, ..., n\}$ と, 各 $i \in I$ に対し m_i 種類のモードが与えられる. 各長方形 $i \in I$ の各モード k ($k = 1, 2, ..., m_i$) について,

$w_i^{(k)}$: 長方形 i のモード k での幅,
$h_i^{(k)}$: 長方形 i のモード k での高さ,
$p_i^{(k)}(x)$: 長方形 i のモード k での x 軸コスト関数,
$q_i^{(k)}(y)$: 長方形 i のモード k での y 軸コスト関数,

が与えられる. 配置は, 各長方形について1つのモードを選択し, さらに x と y の座標値を与えることで定まる. 配置 π における各長方形のモードを $\mu(\pi) = (\mu_1(\pi), \mu_2(\pi), ..., \mu_n(\pi))$ とする. また, 長方形 i の左下隅の座標を $(x_i(\pi), y_i(\pi))$ と記し, x 軸コストの最大値と y 軸コストの最大値を

$$p_{\max}(\pi) = \max_{i \in I} p_i^{(\mu_i(\pi))}(x_i(\pi)), \quad (1)$$

$$q_{\max}(\pi) = \max_{i \in I} q_i^{(\mu_i(\pi))}(y_i(\pi)), \quad (2)$$

と定義する. このとき, 全長方形を二次元平面上に互いに重なりなく配置し, 目的関数

$$g(p_{\max}(\pi), q_{\max}(\pi)) + c(\mu(\pi))$$

を最小化する問題を考える. ただし, 関数 g は, $p_{\max}(\pi)$, $q_{\max}(\pi)$ に関して単調非減少であると仮定する. 本研究で提案するアルゴリズムは, 配置コスト関数 $p_i^{(k)}(x)$ と $q_i^{(k)}(y)$ として任意の区分線形関数 (不連続, 非凸でもよい) を扱うことができるため, 非常に汎用的である. また, モードの導入により, たとえば長方形の90°回転などが実現できるようになっており, さらに汎用性が高くなっている.

3　解の表現方法

本研究では, 順列対 (sequence-pair) 表現 [1] を用いて解を表現する. 順列対表現では, n 個の長方形の順列の対 $\sigma = (\sigma_+, \sigma_-)$ を考える. ここで, $\sigma_+(k) = i$ は, 順列 σ_+ において k 番目の長方形が i であることを意味する (σ_- も同様). $\sigma = (\sigma_+, \sigma_-)$ より二項関係 $\preceq_\sigma^x$ と $\preceq_\sigma^y$ を,

$$\sigma_+^{-1}(i) \leq \sigma_+^{-1}(j) \quad \text{かつ} \quad \sigma_-^{-1}(i) \leq \sigma_-^{-1}(j) \Longleftrightarrow i \preceq_\sigma^x j,$$
$$\sigma_+^{-1}(i) \geq \sigma_+^{-1}(j) \quad \text{かつ} \quad \sigma_-^{-1}(i) \leq \sigma_-^{-1}(j) \Longleftrightarrow i \preceq_\sigma^y j,$$

と定義する. 二項関係 $\preceq_\sigma^x$ と $\preceq_\sigma^y$ は, 定義よりそれぞれ半順序関係になる. また, 任意の対 i と j に対し $i \preceq_\sigma^x j$ (あるいは $j \preceq_\sigma^x i$) と $i \preceq_\sigma^y j$ (あるいは $j \preceq_\sigma^y i$) のちょうど一方が成立するという性質がある. 与えられた $\sigma = (\sigma_+, \sigma_-)$ と $\mu = (\mu_1, \mu_2, ..., \mu_n)$ に対し

- $\mu_i(\pi) = \mu_i$,
- $i \preceq_\sigma^x j$ かつ $i \neq j \Longrightarrow x_i(\pi) + w_i^{(\mu_i)} \leq x_j(\pi)$,
- $i \preceq_\sigma^y j$ かつ $i \neq j \Longrightarrow y_i(\pi) + h_i^{(\mu_i)} \leq y_j(\pi)$,

を満たす配置 π すべての集合を $\Pi_{\sigma,\mu}$ と定義する. このとき, 「任意の σ と μ に対し, $\Pi_{\sigma,\mu} \neq \emptyset$」かつ「任意の配置 π に対し, $\pi \in \Pi_{\sigma,\mu}$ となる (σ, μ) が存在する」という性質が成り立つ.

4　動的計画法

(σ, μ) が与えられたとき, 目的関数を最小にする配置 $\pi \in \Pi_{\sigma,\mu}$ を決定する問題を考え, 動的計画法に基づく多項式時間アルゴリズムを与える. なお, $\preceq_\sigma^x$ と $\preceq_\sigma^y$ の性質, および目的関数に関する仮定から, $p_{\max}(\pi)$ と $q_{\max}(\pi)$ をそれぞれ独立に最小化すれば目的関数を最小化できることが示せるので, ここでは, $p_{\max}(\pi)$ を最小化するアルゴリズムのみを示す. $q_{\max}(\pi)$ についても同様である.

まず, すべての長方形 $i \in I$ について,

$J_i = \{j \in I \mid j \preceq_\sigma^x i\}$,

$f_i(x)$: $\forall j \in J_i$ に対して $x_j(\pi) + w_j^{(\mu_j(\pi))} \leq x$ を満たす $\pi \in \Pi_{\sigma,\mu}$ の中での $\max_{j \in J_i} p_j^{(\mu_j(\pi))}(x_j(\pi))$ の最小値,

と定義する. このとき, $f_i(x)$ は, 漸化式

$$f_i(x) = \min_{t \leq x - w_i^{(\mu_i(\pi))}} \max\{p_i^{(\mu_i(\pi))}(t), \max_{j \in J_i \setminus \{i\}} f_j(t)\}, \quad (3)$$

により計算される.(ただし, $J_i \setminus \{i\} = \emptyset$ のとき, すべての t について $\max_{j \in J_i \setminus \{i\}} f_j(t) = -\infty$ とする.) $p_{\max}(\pi)$ および各長方形の最適な x 座標は, $f_i(x)$ を用いて効率よく求めることができる. 本研究では, (3) の計算を効率良く行うため, データ構造に平衡二分探索木を用いた工夫も加えている. 詳細は略すが, 任意の区分線形関数 $p_i^{(k)}(x)$ に対して, 時間量が低次の多項式であることが示せる. とくに, $p_i^{(k)}(x)$ が単調非減少であるときなど, 実用上重要ないくつかの特殊ケースに対しては, $O(n \log n)$ 時間である. 本研究では, さらに, 順列対から配置を求めるもう一つのアルゴリズム (計算時間は同等だが精度の向上が期待できる) と, 配置から順列対を求める $O(n \log n)$ 時間アルゴリズムも提案し, 局所探索に組み込んでいる.

5 局所探索法

局所探索法は, 現在の解 (σ, μ) の近傍 $N(\sigma, \mu)$ 内に (σ, μ) より良い解があればそれに置き換える, という操作を可能な限り反復する方法である. 通常, 局所探索を 1 度行っただけでは, 未探索の領域にさらに良い解が隠れているという危惧が残るため, 本研究ではメタ戦略の中から, 反復局所探索法 (ILS 法) を試みている. ILS 法は多数の初期解それぞれに局所探索を適用し, 得られた最良の解を出力するものであるが, 初期解の生成に過去の探索で得られたよい解を利用する点に特徴がある.

5.1 クリティカルパス

以下,x 軸方向についてのみ定義するが, y 軸方向も同様である. 配置 $\pi \in \Pi_{\sigma,\mu}$ に対して, 有向グラフ $G = (V, E)$ および長方形の部分集合 $S, T, \tilde{S}, \tilde{T}$ を,

$V = I$,

$(i,j) \in E \Longleftrightarrow x_i(\pi) + w_i^{(\mu_i(\pi))} = x_j(\pi)$ かつ $i \preceq_\sigma^x j$,

$S = \{i \in I \mid p_i(x_i(\pi) - \varepsilon) \geq p_{\max}(\pi)\}$,

$\tilde{S} = \{i \in S \mid p_i(x_i(\pi)) = p_{\max}(\pi)\}$,

$T = \{i \in I \mid p_i(x_i(\pi) + \varepsilon) \geq p_{\max}(\pi)\}$,

$\tilde{T} = \{i \in T \mid p_i(x_i(\pi)) = p_{\max}(\pi)\}$,

とする (ただし ε は十分小さな任意の正数). このとき, 始点を $s \in S$, 終点を $t \in T$ とし, $s \in \tilde{S}$ もしくは $t \in \tilde{T}$ が成り立つ G 上の有向パスをクリティカルパスと定義する. 前節で提案したアルゴリズムによって得られる配置においては, 必ず 1 本以上のクリティカルパスが存在し, これをこわさない限り $p_{\max}(\pi)$ の, すなわち目的関数値の改善は望めない.

5.2 近傍

局所探索を行うための近傍には, 交換近傍, シフト近傍, 限定双シフト近傍, モード変更近傍の 4 つを組み合わせて用いている. 交換近傍とは, σ_+ と σ_- の一方, もしくは両方の順列において, 二つの長方形の位置を互いに交換することで得られる解の集合である. シフト近傍は, σ_+ と σ_- の一方, もしくは両方の順列において, 一つの長方形の位置を移動することで得られる解の集合である. これらの近傍のサイズを効果的に縮小するため, クリティカルパスを用いている. 限定双シフト近傍は, クリティカルパスを壊しながら, 他の長方形の配置になるべく影響を与えないような操作で得られる解の集合である (正確な定義は省略する). モード変更近傍とは, 一つの長方形のモードを変更することで得られる解の集合である.

6 計算実験

我々のアルゴリズムの性能を計算実験により確認した. 用いた問題例は, 配置された全ての長方形を覆う長方形領域の最小化を目的とした長方形詰込み問題で, 長方形数 49 から 500 までの 5 問を調べた. この問題の専用アルゴリズム Seq-Pair[1] と BSG[2], および我々のアルゴリズム Our-ILS を比較した. 表 1 に結果を示す. この結果, 我々のアルゴリズムは汎用的であるにも関わらず, 専用アルゴリズムと比較してさほど劣らない性能であることが確認できる. したがって, 従来の専用アルゴリズムでは処理できない幅広い問題に対して, 有効に利用できることが期待される.

表 1: 提案手法と他のアルゴリズムの比較

size	Seq-Pair		BSG		Our-ILS	
49	96.29	174	97.10	69	96.05	200
100	88.54	248.7	97.08	68.2	95.53	300
146	94.42	678.7	94.87	100.2	95.31	500
200	—	—	—	—	94.69	700
500	90.82	7802.9	94.10	334.6	93.36	2000

左：充填率 (%), 右：計算時間 (秒)

参考文献

[1] H. Murata, K. Fujiyoshi, S. Nakatake and Y. Kajitani, "VLSI Module Placement Based on Rectangle-Packing by the Sequence-Pair," *IEEE Transactions on CAD*, 15, pp.1518-1524, 1996.

[2] S.Nakatake, K.Fujiyoshi, H. Murata and Y. Kajitani, "Module Placement on BSG-Structure and IC Layout Applications," *Proceedings of International Conference on CAD*, pp.484-491, 1996.

[3] T. Takahashi, "An Algorithm for Finding a Maximum-Weight Decreasing Sequence in a Permutation, Motivated by Rectangle Packing Problem (in Japanese)," *Technical Report of IEICE*, VLD96-30, pp.31-35, 1996.