2002年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の高速化

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,NP困難であることが知られている.文献[1]では,この問題に一般的な配置コストを導入し,局所探索法に基づくアルゴリズムを提案した.[1]の問題は,様々な配置問題やスケジューリング問題を自然に定式化できる,非常に汎用性の高い問題である.また,[1]のアルゴリズムの特徴として,順列対表現[2]を用いて解を表現し,低次の多項式時間で順列対から配置を求めるという点が挙げられる.

本研究では,文献[1]のアルゴリズムをもとに,局所探索において近傍内の解を順序よく評価することによって,アルゴリズムの高速化を実現する.様々な問題例に対して計算実験を行い,従来手法との性能比較を行った.

2　問題の定義

長方形集合 $I=\{1,2,\ldots,n\}$ 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 $k\,(k=1,2,\ldots,m_i)$ 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 $\mu(\pi)=(\mu_1(\pi),\mu_2(\pi),\ldots,\mu_n(\pi))$ 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

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

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

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[2]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,

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

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 $j(i\ne j)$ 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 $\mu=(\mu_1,\mu_2,\ldots,\mu_n)$ に対し

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

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,$\Pi_{\sigma,\mu}\ne\emptyset$」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.

© 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.

4 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) とタブー探索法 (TS 法) を試みている.

4.1 近傍

局所探索を行うための近傍には, シフト近傍を用いている. 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. 本研究では, 移動する長方形の選択の際にクリティカルパス [1] (詳細な定義は略すが, これをこわさない限り解の改善が望めないというものである) を利用することで探索の効率化を行っている.

5 解の評価方法

(σ,μ) が与えられたとき, $\Pi_{\sigma,\mu}$ の中で目的関数を最小にする配置 π を求めるアルゴリズムを文献 [1] で提案した. このアルゴリズムは動的計画法に基づいており, 低次の多項式時間で目的関数値および最適配置を計算することができる. とくに, 配置コスト関数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, 計算時間は $O(n\log n)$ である.

5.1 高速化

シフト近傍において, 移動する長方形を決定し, それを両方の順列において移動する場合, 調べる解の個数は $O(n^2)$ となる. これらの解を以下の手法を用いて評価する. ここでは $p_{\max}(\pi)$ の最小値を求めるアルゴリズムのみを示すが, $q_{\max}(\pi)$ についても同様である.

まず, 移動する長方形 (i とする) を取り除いた $n-1$ 個の長方形の集合を $\tilde{I}$, 順列対を $\tilde{\sigma}$ とし, すべての $1 \le p,q \le n$ について,

$$J^f_{p,q} = \{j \in \tilde{I} \mid \tilde{\sigma}_+(j) < p, \tilde{\sigma}_-(j) < q\},$$

$$J^b_{p,q} = \{j \in \tilde{I} \mid \tilde{\sigma}_+(j) \ge p, \tilde{\sigma}_-(j) \ge q\},$$

$f_{p,q}(x)$: $\forall j \in J^f_{p,q}$ に対して $x_j(\pi) + w_j \le x$ を満たす $\pi \in \Pi_{\tilde{\sigma},\mu}$ の中での $\max_{j\in J^f_{p,q}} p_j(x_j(\pi))$ の最小値,

$b_{p,q}(x)$: $\forall j \in J^b_{p,q}$ に対して $x_j(\pi) \ge x$ を満たす $\pi \in \Pi_{\tilde{\sigma},\mu}$ の中での $\max_{j\in J^b_{p,q}} p_j(x_j(\pi))$ の最小値,

と定義する. このとき, $f_{p,q}(x)$ は, p または q が1のとき0となり, そうでないときは漸化式

$$f_{p,q}(x) = \begin{cases} \max\{f_{p-1,q}(x), f_{p,q-1}(x)\}, \\ \quad \sigma_+^{-1}(p-1) \ne \sigma_-^{-1}(q-1) \text{ のとき} \\ \min_{t\le x-w_j}\{\max(p_j(t), f_{p-1,q-1}(t))\}, \\ \quad \sigma_+^{-1}(p-1) = \sigma_-^{-1}(q-1) = j \text{ のとき} \end{cases}$$

により計算される. ($b_{p,q}(x)$ の計算もほぼ同様に行うことができる.) また, $(\tilde{\sigma},\mu)$ に対する x 軸コストの最大値を最小化したものを $p_{\max}(\tilde{\pi})$ と書き, 文献 [1] のアルゴリズムを用いて計算しておく. これらを用いて, 長方形 i を順列 $\tilde{\sigma}_+$ の α 番目, $\tilde{\sigma}_-$ の β 番目に挿入したときの $p_{\max}(\pi)$ の最小値は,

$$\max\{p_{\max}(\tilde{\pi}), \min_t \max(f_{\alpha,\beta}(t), p_i(t), b_{\alpha,\beta}(t+w_i))\}$$

と計算することができる.

この手法を用いると, 文献 [1] のアルゴリズムで解を1つ評価するときの $n/\log n$ 倍の時間で $O(n^2)$ 個の解の評価が可能となり, 解一つあたりの評価時間が $1/n\log n$ 倍となる. とくに, 前述の特殊ケースに対しては, 全体の計算時間が $O(n^2)$, すなわち解一つあたりの計算時間が $O(1)$ となる.

長方形を選択した後, それを一方の順列で移動する場合は, 調べる解の個数が $O(n)$ となるが, これらの解の探索の際は, 文献 [1] のアルゴリズムをうまく使うことで, 全ての p,q に対し $f_{p,q}(x), b_{p,q}(x)$ を計算するのではなく, 必要最小限のもののみ計算するという効率化を行っている. この工夫によって, 文献 [1] のアルゴリズムで解を1つ評価するのと同じオーダの計算時間で, $O(n)$ 個の解全てを評価している.

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と, 各仕事を指定納期の近くに配置することを求めるスケジューリング問題である. 実験結果は発表当日に報告する.

参考文献

[1] S. Imahori, M. Yagiura and T. Ibaraki, "Local Search Algorithms for the Rectangle Packing Problem with General Spatial Costs," submitted for publication.

[2] 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.

© 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.