線形順序付け問題の解法

櫻庭　セルソ　智，柳浦　睦憲

線形順序付け問題は，正方行列の行と列を同じ順列で並べ替え，下三角部分の値の合計を最小化する問題であり，古くから研究されている．この問題は，辺に重みのついた有向グラフから一部の辺を取り除いてすべての有向閉路を除去するとき，取り除いた辺の重みの合計を最小化するフィードバック辺集合問題と等価である．本稿では，この問題に対するさまざまな解法を，実用的解法を中心に紹介する．

キーワード：線形順序付け問題，フィードバック辺集合問題，分枝限定法，局所探索法，メタ戦略

1. はじめに

投票により複数の候補者に順位を与えることを考えてみよう．各投票者がすべての候補者を見比べてから投票するのが容易な場合は，良いと思う候補に投票する方法が通常よく用いられる．しかし，たとえば多くの研究発表のなかから受賞者を決める場合は，一部の発表を聞き逃してしまう投票者がいることがしばしばである．そのような場合に通常の投票方法を採ってしまうと，たまたま多くの人が聞いていた発表は，そうでない発表に比べて有利になってしまう．

そこで，以下のような投票モデルを考えてみよう．各投票者は候補者ではなくいくつかの候補者のペアの各々に対してどちらの候補がより良いと思うかを投票する．そして，できるだけ多くの投票者の意志に合うようにすべての候補者を順位付けする．このような順位付けの問題を定式化すると線形順序付け問題になる．

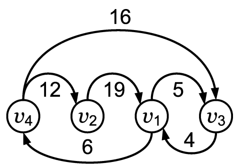
線形順序付け問題（linear ordering problem, LOP）は，正方行列が与えられたとき，行列の行と列を同じ順列で並べ替え，下三角部分の値の合計を最小化する問題である．なお，一般性を失うことなく対角成分は0であると仮定する．行列のすべての要素の重みの合計は常に一定であるので，上三角部分の重み和の最大化も等価な問題である．上の投票の例は，候補者 i のほうが j よりも良いと投票した人の数を行列の第 (i,j) 要素とした場合に対応する．この問題は，経済における産業連関表（industry transaction table）あるいは投入産出表（input-output matrix，I/O 表）[25] と呼ばれるデータの解析に応用があり，古くから研究されている [11]．I/O 表にはある産業の製品がほかの産業の生産活動にどれだけ利用されているかが記されており，経済活動における需給関係の分析に利用されている．なお，Leontief はこのような投入産出分析（input-output analysis）に関する業績により 1973 年にノーベル経済学賞を受賞している．この他にも考古学における出土品の年代順の推定 [16] やスポーツにおけるチームのランキング [17][37] などに応用がある．

線形順序付け問題は有向グラフを用いて定義することもできる．すなわち，辺に重みのついた有向グラフが与えられ，頂点を左から右に 1 列に並べるとき，右から左に向く辺の重みの合計を最小化する問題ととらえるのである．行列の (i,j) 要素を辺 (v_i, v_j) の重みとみなすと，行と列を頂点と同じ順序に並べ替えた行列における下三角部分の重みの合計は，右から左に向く辺の重みの合計に等しい．図 1 に行列による表現とグラフによる表現の例を示す．各頂点 v_i は行列の行と列の番号 i に対応し，下三角部分の数値は右から左に向く辺（図 1（b）の頂点の下側の辺）の重みに対応している．

以下ではまず線形順序付け問題の定式化を行い，計算の複雑さについて述べたのち，分枝限定法，局所探索法，メタ戦略等の実用的解法を紹介する．また，代

	4	2	1	3
4	0	12	0	16
2	0	0	19	0
1	6	0	0	5
3	0	0	4	0

(a)



(b)

図 1　LOP のある解の行列表現 (a) とグラフ表現 (b)

さくらば　せるそ　さとし
Department of Production Engineering, Federal University of Sergipe, Av. Mal. Rondon, S/N, Jardim Rosa Elze, São Cristóvão/SE 49000-000 Brazil
やぎうら　むつのり
名古屋大学大学院情報科学研究科
〒 464–8601 名古屋市千種区不老町

Copyright ⓒ by ORSJ. Unauthorized reproduction of this article is prohibited.

表的なベンチマーク問題例についても簡単に紹介する．

2. 定式化と計算の複雑さ

本節ではまず線形順序付け問題の定式化を与える．本稿ではとくに断らない限り以下のグラフによる最小化問題としての定式化を用いて説明を行う．頂点集合 V $(|V| = n)$ と辺集合 $E \subseteq V \times V$ よりなる有向グラフ $G = (V, E)$，および各辺 $(u, v) \in E$ の重み c_{uv} が与えられる．このとき，頂点を左から右へ 1 列に並べ，右から左へ向く辺の重み和を最小化するのがこの問題の目的である．便宜上，右から左へ向く辺を逆向辺，左から右へ向く辺を順向辺と呼ぶ．また，$(u, v) \notin E$ ならば $c_{uv} = 0$ であると仮定する．頂点の並べ方を順列 $\pi : \{1, \ldots, n\} \to V$ で表す．$\pi(i) = u$ は左から i 番目の頂点が u であることを意味する．これらを用いると，逆向辺の重み和は

$$cost(\pi) = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} c_{\pi(j)\pi(i)}$$

と表せる．これを以下ではコストと呼ぶ．なお，重みが均一な（つまりすべての辺 $(u, v) \in E$ の重み c_{uv} が 1 である）場合，この問題は Slater の問題とも呼ばれ，得られる順位付けを Slater ランキングと呼ぶこともある [35]．このように辺重みが均一である問題例に限定した場合でも線形順序付け問題は NP 困難であることが知られている（後述する等価な問題であるフィードバック辺集合問題の辺重みが均一な場合に対して 1972 年に証明されている [22]）．

一般性を失うことなく G を無向グラフとみなしたときに連結であることを仮定する（この仮定より $m = |E|$ に対して $m \geq n - 1$ である）．また，$c_{uv} \geq c_{vu}$ ならば，$c_{uv} - c_{vu}$ をあらためて辺 (u, v) の重みとし，辺 (v, u) を消去しても問題の本質は変わらない．とくに，$c_{uv} = c_{vu}$ ならば辺 (u, v) と (v, u) の両方を消去できる．これらのことから，すべての辺 $(u, v) \in E$ に対して $c_{uv} > 0$ であると仮定しても一般性を失わない．任意の頂点対 $u, v \in V$ に対して (u, v) と (v, u) のちょうど一方が E に含まれるようなグラフをトーナメント (tournament) と呼ぶが，上述の議論より，（重み 0 の辺を許せば）グラフをトーナメントに限定しても一般性を失わないことがわかる．とくに，辺重みが均一でグラフが一般である場合，すなわちすべての辺の重みを 1 に限定しグラフの形状に制限を設けない場合は，辺重みを 0 か 1 に限定したトーナメント上の問題例に帰着できる．一方，重み 0 の辺を許さない場合はこのような簡単な変換が直ちには利用できないので注意が必要である．実際，グラフをトーナメントに限定した辺重みの均一な線形順序付け問題の計算の複雑さは自明ではなく，1992 年に NP 困難であることが予想されていたものの [2]，10 年以上未解決であった．この予想が肯定的に解決されたのは比較的最近である [1]．

線形順序付け問題は以下のように整数計画問題として定式化することができる：

$$\begin{array}{lll}
\text{最小化} & \displaystyle\sum_{u \in V} \sum_{\substack{v \in V \\ v \neq u}} c_{uv} x_{uv} & \\
\text{制約} & x_{uv} + x_{vu} = 1, & \forall u \neq v \in V, \\
& x_{uv} + x_{vw} + x_{wu} \geq 1, & \forall u \neq v \neq w \in V, \\
& x_{uv} \in \{0, 1\}, & \forall u \neq v \in V.
\end{array}$$

なお，$\forall u \neq v \neq w \in V$ は「すべての相異なる u, v, w に対して」を意味するものとする．$x_{uv} = 1$ は辺 (u, v) が逆向辺であることを表す．最初の制約は任意の頂点対に対して左右を指定しなければならないことを表し，次の制約はそのような左右の指定に推移性があり，全順序が得られることを表す．$x_{uv} = 1$ がグラフから辺 (u, v) を取り除くことを意味するととらえ，長さ 2 の閉路からちょうど 1 本の辺を，長さ 3 の閉路から 1 本以上の辺を取り除かなければならないことを表す制約であると解釈することもできる．その結果グラフが非閉路的になることを容易に示せる．

このように有向グラフから一部の辺を取り除いてすべての有向閉路を除去するとき，取り除いた辺の重みの合計を最小化する問題は，（有向）フィードバック辺集合問題 ((directed) feedback edge set problem, feedback arc set problem) と呼ばれ，古くから多くの研究がある．この問題が線形順序付け問題と等価であることは，取り除く辺集合と，右から左に向く辺の集合が対応することから直ちにわかる．なお，除去する辺の総重みの最小化の代わりに，残す辺の総重みの最大化を考える場合は，最大非閉路部分グラフ問題（maximum acyclic subgraph problem）と呼ばれる．

逆向辺の重み和の最小化を考える場合，この問題は APX 困難であることが示されており [21]，$\mathrm{P} \neq \mathrm{NP}$ の仮定の下では，ある定数 α が存在して，最適値の α 倍以下の解を得ることを保証できる多項式時間近似アルゴリズムは存在しないことが明らかにされている．そのような α の値が 1.3606 以上であることも示されている [13]．一方，最適値の $O(\log n \log \log n)$ 倍以下の解を得る保証のある多項式時間近似アルゴリズムも提

 Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.

案されている [14][34].

順向辺の重み和の最大化を考える場合は，簡単な方法で最適値の 1/2 以上の重みをもつ解を得ることができる．適当な順列 π とそれを逆順にした順列 π' を考えると，これらの順向辺の集合は互いに疎であり，しかもそれらの和集合はすべての辺集合となる．最適値はすべての辺の重み和以下であるので，π と π' のうち重み和の大きいほうを取れば，最適値の 1/2 以上となる．長さ 2 の閉路がないグラフで辺重みが均一な場合については，グラフの最大次数 Δ に対して，順向辺の数が $(1/2+\Omega(1/\sqrt{\Delta}))m$ 以上となる順列を得るアルゴリズムが提案されている [4]．近似精度保証についてはその他にもさまざまな結果がある [10].

グラフが特殊な構造をもつ場合に対する多項式時間アルゴリズムもいくつか提案されている．以下はいずれも重みが均一な場合に対する結果である．たとえばグラフが特殊な階層構造（定義はやや複雑であるので割愛する）をもつ場合に対しては，線形時間のアルゴリズムがある [12]．また，トーナメントが cyclic と呼ばれる規則的な構造をもつときは，最適解が容易に得られることが知られている [24].

3. 厳密解法

厳密な最適解を得る代表的な手法である分枝限定法（branch-and-bound method）に基づくアルゴリズムの研究は古くから行われている [20]．分枝限定法は，問題をいくつかの小規模な部分問題に分割し，そのすべてを解くことで等価的に元の問題を解くという考え方に基づいている．その際，ある部分問題が元の問題の最適解を与えないことを結論づける手法を組み合わせることで，探索の効率を上げるところに特徴がある．そのような手法の代表例が以下の下界値テストである．発見的解法や分枝限定法の探索中に得られた解 π の中で $cost(\pi)$ が最小のものを暫定解と呼び，π^* と記す．ある部分問題に対して

$$\text{その部分問題の最適値の下界値} \geq cost(\pi^*)$$

であれば，その部分問題を厳密に解いても π^* よりも良い解は得られないため，その部分問題をそれ以上分割して調べる必要のないことが結論できる．このように部分問題の探索を打ち切ることを終端するという．

線形順序付け問題に対する下界の計算方法の例を一つ紹介しよう．有向閉路の集合で，どの二つの閉路をとっても有向辺を共有しないものを見つける．各閉路の中で最小の辺重みを求め，すべての閉路に対してそれらの和をとると，最適値の下界となる（文献 [10] の Theorem 37）．このアイデアに基づいて下界を最大にするような閉路の組合せを求めることは容易ではないが，最小の辺重みのなるべく大きい閉路を選んで組み合わせることで比較的よい下界が得られる．なお，このような閉路に基づく下界の有効性はほかの問題（MAX-2-SAT）に対しても確認されている [19].

線形順序付け問題に対する分枝限定法の比較的新しい成果としては文献 [9] がある．この文献では，メタ戦略の一種である評価関数摂動法（noising method）と呼ばれる発見的解法により初期暫定解を得，ラグランジュ緩和を用いて下界を得る方法を提案している．緩和問題は元の問題に比べると高速に解けるものの，分枝限定法においては下界値計算の回数が通常非常に多く，その都度緩和問題を解くのには時間がかかるので，緩和問題を解くことなく部分問題を終端する工夫も行っている．ランダムに生成した問題例に対する計算実験の結果，頂点数 100 までの問題例でワークステーション Sun SPARC を用いて約 1 時間以内に解けるものが多数あること，および問題の難しさは頂点数だけではなく辺重みなどの生成方法に大きく依存することを報告している．なお，このアルゴリズムを実装したプログラムが文献 [9] の著者のウェブサイト[1]で公開されている．

線形順序付け問題に対しては，切除平面法（cutting plane method）に基づくアルゴリズムも提案されている．整数計画問題のすべての実行可能解が満たすような線形不等式制約を妥当不等式と呼ぶが，切除平面法は，線形緩和（linear relaxation, LP relaxation）問題の最適解が満たしていないような妥当不等式がある限りその一部を逐次的に追加しつつ線形緩和問題を解くという操作を反復する方法である．たとえば，文献 [27] では主双対内点法（primal-dual interior point method）を用いた切除平面法が提案されている．また，文献 [28] では，切除平面法において，前半の反復で主双対内点法を用い，後半ではシンプレックス法を用いる方法を提案し，主双対内点法とシンプレックス法の一方のみを用いる方法に比べて大きな改善が得られることを報告している．ランダムに生成した頂点数 100 から 250 の問題例に対する計算実験の結果，これらのすべてが Sun SPARC 20/71 上で 30 分以内に解けたことを報告している．

分枝限定法に切除平面のアイデアを追加した分枝カッ

[1] http://perso.telecom-paristech.fr/~charon/tournament/median.html

Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.

ト法（branch-and-cut method）が有効であることはさまざまな問題に対して確認されているが，線形順序付け問題に対してもそのようなアイデアの有効性が検証されている．たとえば文献 [17] ではさまざまな妥当不等式を利用した切除平面法に分枝限定法を組み合わせる手法を提案しており，頂点数 60 までの問題例が IBM 370/168 を用いて十数分以下で解けることを報告している．また，文献 [30] では，mod-k cut と呼ばれる妥当不等式を分枝カット法で利用することの効果が検証されている．

2 節の整数計画問題による定式化を用いて商用の整数計画ソルバーを利用することで厳密な最適解を求めることもできる．そのようなソルバーの中で代表的なものの一つである CPLEX 10.0 を用いて Xeon（2.8 GHz）を搭載した計算機上でベンチマーク問題例（7 節に後述）のいくつかを解いてみたところ，75 頂点以下の問題例であれば厳密な最適解を得ることができた．一方，頂点数 100 以上の問題例に対しては，12 時間かけても最適解を得ることはできなかった．

4. 発見的解法

線形順序付け問題に対しては多数の発見的解法が提案されている．本節では構築型解法等の比較的単純なアルゴリズムをいくつか紹介する．まず，単純な構築型解法として，簡単なスコアに基づいて頂点を並べる方法がある．たとえば，各頂点 u から出る辺の重みの合計 $\sum_{v\in V} c_{uv}$ をその頂点のスコアとすると，スコアの大きい頂点ほど左に並べるほうが得であることが直感的に理解できるであろう．このような直感に基づいて，スコアの大きい順に左から右へと頂点を並べるのである．このような方法は自然であり，18 世紀後半に Borda [5] によって提案されたとの記述がある [10]．スコアとして，各頂点 u から出る辺の重みの合計 $\sum_{v\in V} c_{uv}$ を u に入る辺の重みの合計 $\sum_{v\in V} c_{vu}$ で割った比，あるいはこれらの差を用いる方法もある．

スコアの最も大きい頂点を一番左に置き，その頂点を取り除いたグラフにおいて再度スコアを計算し直して，修正したスコアの最も大きい頂点を 2 番目に置く，というようにスコアを修正しつつ順列を構築していく方法もある．

フィードバック辺集合問題において「取り除く」辺集合を「向きを反転する」辺集合ととらえ，そのような辺集合を逐次的に選んでいく方法もある．すなわち，グラフから有向閉路がなくなるまで逐次的に辺の向きを反転させる操作を繰り返すのである．辺の重みが一般の場合，グラフをトーナメントに限定しても一般性を失わないことは前述のとおりである．頂点 u から出る辺の本数すなわち出次数を $\delta_+(u)$ と記す．トーナメントに $\delta_+(u) \le \delta_+(v)$ であるような辺 (u,v) があれば，(u,v) を含む長さ 3 の有向閉路が少なくとも一つ存在する [29]．また，そのような辺 (u,v) の向きを反転することで，長さ 3 の有向閉路の数を $\delta_+(v)-\delta_+(u)+1$ 減らすことができる．したがって，そのような辺を反転する操作を繰り返すことで，最終的に閉路を含まないトーナメントに変形できるのである．

反転する辺を選ぶ基準にはいろいろあるが，たとえば辺重みが均一な場合に対しては，$\delta_+(v)-\delta_+(u)+1$ を最大にする辺 (u,v) を選ぶ方法 [36] が提案されている．また，辺重みが一般の場合については，$(\delta_+(v)-\delta_+(u)+1)/c_{uv}$ を最大にする辺 (u,v) を選ぶ方法や，(u,v) を含む長さ 3 の閉路の重み和と (u,v) を反転させたときに現れる長さ 3 の閉路の重み和の差が大きい辺を選ぶ方法などが考案されている [3]．

構築型の解法を反復することで解を改善していく方法も提案されている [7]．この方法では，Insert，Sort，Reverse の 3 つの操作を反復する．まず，$\textsc{Insert}(u,(\pi(1),\dots,\pi(i)))$ は，頂点 u を部分的な順列 $(\pi(1),\dots,\pi(i))$ のいずれかの場所に挿入した順序 $(u,\pi(1),\dots,\pi(i))$，$(\pi(1),u,\pi(2),\dots,\pi(i)),\dots$ のうちコストが最小のものを返す．これを用いて Sort を以下のように再帰的に定義する：$\textsc{Sort}(\pi(1)) = \pi(1)$，$\textsc{Sort}(\pi(1),\dots,\pi(i)) = \textsc{Insert}(\pi(i),\textsc{Sort}(\pi(1),\dots,\pi(i-1)))$ $(i \ge 2)$．つまり Sort は，$i=2,3,\dots$ の順に，Insert を用いて $\pi(i)$ を部分的な順序 $(\pi(1),\dots,\pi(i-1))$ の最良の位置に挿入する操作を繰り返す手続きである．Reverse は与えられた順列の逆向きの順列を返す．また，Sort を適用しても順列に変化が起こらなくなるまで Sort を反復する手続きを Sort$*$ と記す．任意の順列に対して Reverse を適用したのち Sort$*$ を適用することで，操作を適用する前の順列に比べて同等以上の順列が得られることが証明されている．また，この手法により，60 頂点までの LOLIB の問題例に対して相対誤差 2%以内の解が得られることが報告されている．

5. 局所探索法

局所探索法（local search）は解を逐次的に改善していく基本的な手法である．解に少しの変形を加える操作を近傍操作といい，近傍操作によって生成されうる解の集合を近傍と呼ぶ．局所探索法は，適当な初期解

Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.

π_{init} から始め，現在の解 π の近傍内により良い解 π' があれば π を π' に置き換える操作を，近傍内に改善解が存在しなくなるまで反復する方法である．このように近傍内に改善解のない解を局所最適解と呼ぶ．

順列を求める問題に対する代表的な近傍操作としては，以下のものがある．

- 隣接交換: 順列において隣り合う 2 つの頂点の位置を交換する．
- 交換: 2 つの頂点の順列における位置を交換する．
- 挿入: 1 つの頂点を順列のほかの位置に挿入する．

これらの操作によって生成されうる解集合をそれぞれ隣接交換近傍，交換近傍，挿入近傍と呼ぶ．

線形順序付け問題においては，交換操作で改善できる解は挿入操作でも改善できるが，挿入操作で改善できる解を必ずしも交換操作で改善できるとは限らないことが知られている [18]．また，実際に挿入近傍による局所探索法のほうが交換近傍を用いた場合よりも性能がよい傾向にあることが計算実験により観測されている [33]．なお，隣接交換近傍は挿入近傍と交換近傍の両方に含まれる．以上のことから，線形順序付け問題に対するほとんどの局所探索法において，挿入近傍が主要な近傍として用いられている．

近傍内の改善解の 1 つを見つけるか，あるいはそのような解がないと結論づけるまでに要する時間を 1 ラウンド時間（one-round time）という [38]．なお，そのような計算を効率化するためにデータ構造等を工夫することがしばしばあるが，そのような場合には，解の更新に伴って必要となるデータ構造の更新に要する時間も 1 ラウンド時間に含まれる．局所探索法を効率的に実現するためには，この時間を短くすることが重要である．

近傍内の解のコストを評価する際には，通常各解のコストそのものではなく，現在の解からの変化量を計算することで効率化を図る．このように変化量を計算すれば，隣接交換近傍内の 1 つの解のコストを $O(1)$ 時間で計算できることは容易にわかる．隣接交換近傍のサイズは $O(n)$ なので，1 ラウンド時間は $O(n)$ である．一方，挿入近傍と交換近傍については，このように各解のコストの変化量を計算する方法を単純に実装すると，一つの近傍解の評価に $O(n)$ 時間を要する．これらの近傍サイズは $O(n^2)$ なので，1 ラウンド時間は $O(n^3)$ である．挿入近傍については，簡単なアイデアでこれを $O(n^2)$ に改善できる [33]．たとえば，i 番目の頂点を j 番目 $(i < j)$ の頂点の次の位置に挿入する操作を考える．これは，i 番目と $i+1$ 番目の頂点を交換し，次に $i+1$ 番目と $i+2$ 番目の頂点を交換し，というように隣接交換操作を繰り返し，最後に $j-1$ 番目と j 番目の頂点を交換することで実現できる．この途中で出てくるすべての解が挿入近傍内の解であり，その 1 つ 1 つの評価が $O(1)$ 時間でできることから，結局挿入先の候補を近い位置から順序よく探索していくことで，上述の 1 ラウンド時間が実現できるのである．

データ構造を工夫することで，この計算時間をさらに速くする 2 つのアイデアが文献 [32] で提案されている．1 つ目は LIST と呼ばれるアルゴリズムで，各頂点に接続する頂点集合を，現在の解の順に並べたリストを用意することで，1 ラウンド時間 $O(m)$ を実現している．2 つ目は TREE と呼ばれるアルゴリズムで，2–3 木と呼ばれる平衡木に基づいている．各頂点 u に対して 2–3 木を用意し，その葉に u を挿入する場所を対応させ，葉から根へのパス上の点に保持している値の合計によって挿入後のコストを計算できるように情報を管理することで，1 ラウンド時間 $O(n+\Delta\log\Delta)$ を実現している．計算実験の結果，たとえば頂点数 8000 で辺密度 100%（つまり辺数 3000 万以上!）の大規模な問題例であっても，アルゴリズム TREE を用いると，Xeon（NetBurst）3.0 GHz を搭載した計算機で 15 分程度で局所最適解に到達することが確認されている．一方，1 ラウンド時間が $O(n^2)$ 時間のアルゴリズムや LIST を用いた場合は，12 時間かけても局所最適解に到達しない．

次の節で紹介するメタ戦略アルゴリズムの多くが局所探索法に基づいていることから，このような局所探索の効率化はきわめて重要であり，上述の成果は線形順序付け問題に対するメタ戦略の発展に大きく寄与すると考えられる．

6. メタ戦略

構築型の発見的解法や局所探索法などの基本的な解法よりも多少時間はかかってもより良い解を得るために，基本的な解法を高度に組み合わせて設計されたアルゴリズムを総称してメタ戦略（metaheuristics）という．線形順序付け問題に対して提案されているメタ戦略アルゴリズムには，タブー探索法（tabu search）[23]，散布探索法（scatter search）[6]，反復局所探索法（iterated local search）[31]，可変近傍探索法（variable neighborhood search）[15]，遺伝アルゴリズム（genetic algorithm）[8][18] などがある．

文献 [23] のタブー探索法は，1 つの頂点をランダムに選び，その頂点を現在の位置以外の最良の場所に挿

Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.

入する操作を基本ステップとする．一度位置を変更した頂点の移動をしばらくの間禁止することで同じ頂点の位置が頻繁に変更されることを防いでいる．また，各頂点に接続している辺重みの総和をその点の影響力と考え，各反復で移動対象を選ぶ際に影響力の大きいものほど選ばれやすくするフェーズと，各頂点 u に対してその点の位置をこれまでの探索で変更した回数 $\textsc{Freq}(u)$ を用いて $\textsc{Freq}(u)$ が小さいものほど選ばれやすくするフェーズを交互に繰り返すことで，多様な解を探索することを狙っている．さらに，パス再結合（path relinking）と呼ばれる 2 つの解を組み合わせて新たな解を生成する手法を用いて，有望な領域に探索を集中化するなどの工夫も行っている．

過去に得られた良い解にランダムな摂動を与えた解を初期解として局所探索を行う操作を反復する手法を反復局所探索法という．摂動を与える際には，近傍内から解をランダムに選ぶ操作を用いることが多いが，このための近傍として大きさの異なるものを複数用意しておき，探索の状況に応じて摂動の大きさを変える方法を可変近傍探索法と呼ぶ．文献 [15] では，この枠組みの下で局所探索の部分にタブー探索法を用いる方法や，摂動を与える操作において上述の $\textsc{Freq}$ を用いる方法などを検討している．

解の集団を保持し，集団の中のいくつかの解を利用して新たな解を生成したのち，得られた解の中で良いものを残す操作を反復することで解集団を更新していく方法を遺伝アルゴリズムと呼ぶ．このような枠組みに局所探索法を組み合わせた方法は，遺伝的局所探索法などと呼ばれている．線形順序付け問題に対してもこのアイデアに基づくアルゴリズムが提案されており，その効果が確認されている [8][18]．文献 [8] では，さらに，この枠組みにおいて局所探索の代わりに評価関数摂動法やアニーリング法（simulated annealing）と呼ばれるメタ戦略を用いる方法も検証している．

遺伝アルゴリズムと同様に解集団に基づく手法に散布探索法がある．遺伝アルゴリズムは生物の進化の様子に着想を得て提案された手法であるが，散布探索法ではそのような背景にとらわれず，問題構造や探索履歴を利用したルールを導入してアルゴリズムが構成されることが多い．たとえば，遺伝アルゴリズムにおける交叉と呼ばれる操作では 2 つの解を組み合わせて新たな解を生成するが，文献 [6] の散布探索法では，3 つ以上の解からも新たな解を生成するルールを提案している．

上述のメタ戦略アルゴリズムの多くは，頂点数 200 までのベンチマーク問題例に対して性能評価が行われており，高い性能を示すことが確認されている．たとえば，文献 [15] では，可変近傍探索法やタブー探索法は，最適解が既知であるような問題例のほとんどに対して最適解を得ることができ，その計算時間は Pentium 4（2 GHz）を搭載した計算機上で平均 1 秒未満であったことを報告している．文献 [31] では，前節で紹介した TREE アルゴリズムに基づく局所探索法を反復局所探索法に組み込み，頂点数最大 8000 までの大規模な問題例に対する計算実験を通して，その有効性を確認している．

7. ベンチマーク問題例

本節では線形順序付け問題に対する代表的なベンチマーク問題例を紹介しておく．古くから利用されてきた問題例として，LOLIB がある[2]．これらはヨーロッパの投入産出表の現実のデータから得られた 49 の問題例からなり，その頂点数は 44 から 60 である．これらのすべてに対して厳密な最適解が知られている．

同様のデータとして，SGB（Stanford GraphBase）と呼ばれる問題例群がある．これらはアメリカ合衆国の投入産出表から得られた 25 問からなり，その頂点数は 75 である．これらについてもすべての問題例に対して厳密な最適解が知られており，問題例とともに公開されている[3]．このサイトにはこの他にもいくつかの問題例群が公開されている．たとえば，ランダムに生成された頂点数最大 250 までの問題例や，テニスの ATP ワールドツアーの結果から生成された問題例などがある．

線形順序付け問題の問題例を生成するプログラムも公開されており[4]，文献 [28] の計算実験等に利用されている．

著者らのサイトでは，頂点数 500 から 8000 までの大規模な問題例を公開している[5]．これらの問題例は，ランダムにグラフを生成し，各辺の重みを区間 [1, 99] の整数から一様にランダムに選んで生成したものである．辺密度が疎なものから密なものまでさまざまな種類の問題例がある．

8. まとめ

本稿では，線形順序付け問題の基本的な性質と，この

[2] http://www2.iwr.uni-heidelberg.de/groups/comopt/software/LOLIB/
[3] http://www.optsicom.es/lolib/
[4] http://homepages.rpi.edu/~mitchj/generators/linord/
[5] http://www.co.cm.is.nagoya-u.ac.jp/~yagiura/lop/

Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.

問題に対する厳密解法や発見的解法などの実用的な解法を紹介した．この問題に対して提案されているアルゴリズムはもちろん本稿で紹介したものだけではないが，より詳しくは Charon と Hudry による 50 ページ以上にわたる詳しいサーベイ [10] や線形順序付け問題に対する厳密解法や発見的解法の話題を集めた Martí と Reinelt による本 [26] をご参照いただきたい．本稿が線形順序付け問題に対する実用的な解法に関する理解を深める一助となれば幸いである．

謝辞　貴重なコメントをいただいた今堀慎治，小野廣隆，佐々木美裕，野々部宏司，橋本英樹，原口和也の諸氏に謝意を表します．

参考文献

[1] N. Alon: Ranking tournaments. *SIAM Journal on Discrete Mathematics*, **20** (2006), 137–142.

[2] J. Bang-Jensen and C. Thomassen: A polynomial algorithm for the 2-path problem for semicomplete digraphs. *SIAM Journal on Discrete Mathematics*, **5** (1992), 366–376.

[3] J.P. Barthélemy, A. Guénoche and O. Hudry: Median linear orders: Heuristics and a branch and bound algorithm. *European Journal of Operational Research*, **42** (1989), 313–325.

[4] B. Berger and P. Shor: Approximation algorithms for the maximum acyclic subgraph problem. *Proc. 1st ACM-SIAM Symposium on Discrete Algorithms - SODA'90*, (San Francisco, CA, 1990) 236–243.

[5] J.C. Borda: Mémoire sur les élections au scrutin. *Histoire de l'Académie Royale des Sciences pour 1781* (Paris, 1784).

[6] V. Campos, F. Glover, M. Laguna and R. Martí: An experimental evaluation of a scatter search for the linear ordering problem. *Journal of Global Optimization*, **21** (2001), 397–414.

[7] S. Chanas and P. Kobylanski: A new heuristic algorithm solving the linear ordering problem. *Computational Optimization and Applications*, **6** (1996), 191–205.

[8] I. Charon and O. Hudry: Lamarckian genetic algorithms applied to the aggregation of preferences. *Annals of Operations Research*, **80** (1998), 281–297.

[9] I. Charon and O. Hudry: A branch-and-bound algorithm to solve the linear ordering problem for weighted tournaments. *Discrete Applied Mathematics*, **154** (2006), 2097–2116.

[10] I. Charon and O. Hudry: An updated survey on the linear ordering problem for weighted or unweighted tournaments. *Annals of Operations Research*, **175** (2010), 107–158.

[11] H.B. Chenery and T. Watanabe: International comparisons of the structure of production. *Econometrica*, **26** (1958), 487–521.

[12] V. Conitzer: Computing Slater rankings using similarities among candidates. *Proc. 21st National Conference on Artificial Intelligence - AAAI2006*, (Boston, MA, USA, 2006), 613–619.

[13] I. Dinur and S. Safra: On the hardness of approximating minimum vertex cover. *Annals of Mathematics*, **162** (2005), 439–485.

[14] G. Even, J.S. Naor, M. Sudan and B Schieber: Approximating minimum feedback sets and multicuts in directed graphs. *Algorithmica*, **20** (1998), 151–174.

[15] C.G. Garcia, D. Pérez-Brito, V. Campos and R. Martí: Variable neighborhood search for the linear ordering problem. *Computers and Operations Research*, **33** (2006), 3549–3565.

[16] F. Glover, T. Klastorin and D. Klingman: Optimal weighted ancestry relationships. *Management Science*, **20** (1974), 1190–1193.

[17] M. Grötschel, M. Jünger and G. Reinelt: A cutting plane algorithm for the linear ordering problem. *Operations Research*, **32** (1984), 1195–1220.

[18] G. Huang and A. Lim: Designing a hybrid genetic algorithm for the linear ordering problem. *Proc. Genetic and Evolutionary Computation Conference - GECCO 2003*, (Chicago, 2003), 1053–1064.

[19] T. Ibaraki, T. Imamichi, Y. Koga, H. Nagamochi, K. Nonobe and M. Yagiura: Efficient branch-and-bound algorithms for weighted MAX-2-SAT. *Mathematical Programming*, **127** (2011), 297–343.

[20] R. Kaas: A branch and bound algorithm for the acyclic subgraph problem. *European Journal of Operational Research*, **8** (1981), 355–362.

[21] V. Kann: *On the Approximability of NP-complete Optimization Problems*, PhD thesis, KTH Royal Institute of Technology, Sweden, 1992.

[22] R.M. Karp: Reducibility among combinatorial problems. in: R.E. Miller and J.W. Thatcher, eds., *Complexity of Computer Computations* (Plenum Press, New York, 1972) 85–103.

[23] M. Laguna, R. Martí and V. Campos: Intensification and diversification with elite tabu search solutions for the linear ordering problem. *Computers and Operations Research*, **26** (1999), 1217–1230.

[24] J.-F. Laslier: *Tournament Solutions and Majority Voting* (Springer, 1997).

[25] W. Leontief: *Structure of American Economy 1919–1929* (Harvard University Press, Cambridge, 1941).

[26] R. Martí and G. Reinelt: *The Linear Ordering Problem: Exact and Heuristic Methods in Combinatorial Optimization*, Applied Mathematical Sciences, Vol. 175 (Springer, 2011).

[27] J.E. Mitchell and B. Borchers: Solving real world linear ordering problems using a primal-dual interior point cutting plane method. *Annals of Operations Research*, **62** (1996), 253–276.

[28] J.E. Mitchell and B. Borchers: Solving linear ordering problems with a combined interior point/simplex cutting plane algorithm. *High Performance Optimization* (Kluwer, Dordrecht, 2000).

[29] J.W. Moon: *Topics on Tournaments* (Holt, Rinehart and Winston, New York, 1968).

[30] M. Oswald, G. Reinelt and H. Seitz: Applying mod-k-cuts for solving linear ordering problems. *Top*, **17** (2009), 158–170.

[31] C.S. Sakuraba, D.P. Ronconi and M. Yagiura: Iterated local search para problemas de ordenação linear

Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.

de grande porte. *Proc. XLIII Simpósio Brasileiro de Pesquisa Operacional*, (Ubatuba, 2011), 1606–1617.
[32] C.S. Sakuraba and M. Yagiura: Efficient local search algorithms for the linear ordering problem. *International Transactions in Operational Research*, **17** (2010), 711–737.
[33] T. Schiavinotto and T. Stützle: The linear ordering problem: Instances, search space analysis and algorithms. *Journal of Mathematical Modelling and Algorithms*, **3** (2004), 367–402.
[34] P.D. Seymour: Packing directed circuits fractionally. *Combinatorica*, **15** (1995), 281–288.
[35] P. Slater: Inconsistencies in a schedule of paired comparisons. *Biometrika*, **48** (1961), 303–312.
[36] A.F.M. Smith and C.D. Payne: An algorithm for determining Slater's i and all nearest adjoining orders. *British Journal of Mathematical and Statistical Psychology*, **27** (1974), 49–52.
[37] N. Sukegawara, Y. Yamamoto and L. Zhang: Lagrangian relaxation and pegging test for linear ordering problems. *Journal of the Operations Research Society of Japan*, **54** (2011), 142–160.
[38] M. Yagiura and T. Ibaraki: Analyses on the 2 and 3-flip neighborhoods for the MAX SAT. *Journal of Combinatorial Optimization*, **3** (1999), 95–114.

 Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.