日本オペレーションズ・リサーチ学会
2021年 秋季研究発表会

分解定理に基づく線形順序付け問題の発見的解法

01013123　静岡大学　安藤和敏　ANDO Kazutoshi
(入会申請中)　静岡大学　*杉本達哉　SUGIMOTO Tatsuya
02203940　東京理科大学　鮏川矩義　SUKEGAWA Noriyoshi

1. はじめに

V を有限集合とし, $m = |V|$ とする. **線形順序付け問題** (Linear Ordering Problem, LOP) とは, 実数行列 $C = (c_{uv})_{u,v \in V}$ が与えられたとき, 目的関数

$$\sum_{i=1}^{m-1} \sum_{j=i+1}^{m} c_{\sigma_i \sigma_j}$$

を最大化する順列 $\sigma : \{1, \ldots, m\} \to V$ を見出す問題である. 線形順序付け問題は NP 困難な組合せ最適化問題であるが, 投入算出表の三角化や個人の選好の集約などの多くの応用を持つ [3].

一般性を失わず線形順序付け問題の入力行列 $C = (c_{uv})_{u,v \in V}$ は, 非負, かつ, すべての $u, v \in V$ に対して $c_{uv} = 0$ または $c_{vu} = 0$ であると仮定できる. そのような行列 C に対して**強多数グラフ** $G(C) = (V, A(C))$ を,

$$A(C) = \{(u, v) \mid u, v \in V, c_{uv} > 0\}$$

によって定義する. $\mathcal{P}(C)$ を強多数グラフ $G(C)$ の強連結成分に対応する V の分割とする. また, $\mathcal{P}(C)$ の順序付け $(Y_1, \ldots, Y_l)$ は, もし

$$((u, v) \in A(C), u \in Y_i, v \in Y_j, i \neq j) \Longrightarrow i < j$$

が成り立つならば, $\mathcal{P}(C)$ の**位相的順序付け**と呼ばれる. 線形順序付け問題は以下の定理が示すように強多数グラフ $G(C)$ の強連結成分に対応する部分問題に分解できることが知られている.

行列 C を入力とする線形順序付け問題を LOP(C) で表す.

定理 1.1 (分解定理 [2]) 行列 C を線形順序付け問題の入力とし, $(Y_1, \ldots, Y_l)$ を $\mathcal{P}(C)$ の位相的順序付けとする. 各 $i = 1, \ldots, l$ に対して C の Y_i への制限を $C[Y_i]$ とする. LOP($C[Y_i]$) の最適解を $\sigma^{(i)} : \{1, \ldots, |Y_i|\} \to Y_i$ としたとき, 順列 $\sigma^* = (\sigma^{(1)}, \ldots, \sigma^{(l)})$ は LOP(C) の最適解である.

定理 1.1 によって, 入力行列 C に対してもし強多数グラフ $G(C)$ がいくつかのより小さい強連結成分に分解されれば, LOP(C) を解くための各種アルゴリズムの計算時間の短縮が可能になる. しかし, 多くの問題例においては $G(C)$ は強連結であるか, あるいは, $G(C)$ の強連結成分にサイズが大きいものが存在するため, 定理 1.1 によるアルゴリズムの計算時間の短縮は期待できない. 本研究の目的は, 定理 1.1 が問題サイズの縮小に失敗する問題例に対して, 定理 1.1 で与えられるものと類似する分解によって問題サイズを縮小する発見的解法を提案することである. さらに導入した手法の有効性を数値実験によって検証する.

2. 分解定理に基づく発見的解法

$C = (c_{uv})_{u,v \in V}$ を線形順序付け問題の入力とする. 任意の $\gamma \geq 0$ に対して **γ-レベルグラフ** $G_\gamma(C) = (V, A_\gamma(C))$ を,

$$A_\gamma(C) = \{(u, v) \mid u, v \in V, c_{uv} > \gamma\}$$

によって定義する. $G_\gamma(C)$ の強連結成分に対応する V の分割を $\mathcal{P}_\gamma(C)$ とし, $(Z_1, \ldots, Z_l)$ を $\mathcal{P}_\gamma(C)$ の位相的順序付けとする. そして, 定理 1.1 と同様に各 $i = 1, \ldots, l$ に対して LOP($C[Z_i]$) の最適解を $\sigma^{(i)} : \{1, \ldots, |Z_i|\} \to Z_i$ としたとき, V の順列 $\sigma^* = (\sigma^{(1)}, \ldots, \sigma^{(l)})$ は, γ が十分小さければ LOP の良い近似解になることが期待できる. アルゴリズム 1 はこの考えに基づくものである.

線形順序付け問題の入力の行列 $C = (c_{uv})_{u,v \in V}$ に対して, $G(C) = (V, A(C))$ をその強多数グラフとし, $G(C)$ の各枝 (u, v) の重みを c_{uv} によって定義する. 重み付きグラフ $G(C)$ の**カット**とは, $V = V_1 \cup V_2$, $V_1 \cap V_2 = \emptyset$ であるような非空な点部分集合の順序対 (V_1, V_2) である.

$$c(V_1, V_2) = \sum \{c_{uv} \mid (u, v) \in A(C), u \in V_1, v \in V_2\}$$

を最小にする $G(C)$ のカットは**最小カット**と呼ばれる. もう一つの発見的解法は $G(C)$ の誘導部分グラフの最小カット計算に基づくものである. これをアルゴリズム 2 に示す.

3. 予備実験

本論文で導入する二つの発見的解法を Python で実装した. ここで, `MAXSIZE` は 60 に設定し, `MAXSIZE` 以下のサイズの問題を整数計画問題として定式化して PuLP ラ

Def `level`(C,V)
　入力: 行列 $C = (c_{uv})_{u,v \in V}$
　相異なる c_{uv} $(u, v \in V)$ の値を

$$0 = \gamma_0 < \gamma_1 < \cdots < \gamma_k$$

　とする;
　`decom`$(C, V, -1)$;
end
Def `decom`(D, Z, j)
　入力: 行列 $D = (d_{uv})_{u,v \in Z}$, 整数 j $(-1 \leq j \leq k)$
　if $|Z| \leq$ `MAXSIZE` **then**
　　return LOP(D) の厳密解法による解;
　end
　else
　　$(Z_1 \ldots, Z_l)$ を $G_{\gamma_{j+1}}(D)$ の強連結成分分解の位相的順序付けとする;
　　for $i = 1$ to l **do**
　　　$\sigma^{(i)} \leftarrow$ `decom`$(D[Z_i], Z_i, j+1)$;
　　end
　　return $(\sigma^{(1)}, \ldots, \sigma^{(l)})$;
　end
end

アルゴリズム 1: レベルグラフに基づく発見的解法.

Def `mincut`(C,V)
　入力: 行列 $C = (c_{uv})_{u,v \in V}$
　if $|V| \leq$ `MAXSIZE` **then**
　　return LOP(C) の厳密解法による解;
　end
　else
　　$(V_1, V_2) \leftarrow G(C)$ の最小カット;
　　for $i = 1$ to 2 **do**
　　　$\sigma^{(i)} \leftarrow$ `mincut`$(C[V_i], V_i)$;
　　end
　　return $(\sigma^{(2)}, \sigma^{(1)})$;
　end
end

アルゴリズム 2: 最小カットに基づく発見的解法.

表 1: xLOLIB に対する実験結果の一部.

Instance	rel.dev. [%]		Time [s]	
	`level`	`mincut`	`level`	`mincut`
be75eec_150	6.3	4.6	18.7	1083.0
be75np_150	6.3	4.3	348.1	1546.4
be75oi_150	3.0	2.5	15.3	1275.5
be75tot_150	6.6	4.6	35.9	1272.5
stabu1_150	6.8	4.3	127.3	3705.8
stabu2_150	6.3	4.5	878.7	2029.4
stabu3_150	5.7	4.4	235.1	1239.8

イブラリを使用して厳密解を求める．ベンチマーク問題 xLOLIB ($m = 150$) の問題例を入力としてこれらのプログラムを実行した結果の一部を表 1 に示した．表中で rel.dev. は発見的解法による解の目的関数値のこれまでに知られている最良解の目的関数値 ([1]) からの相対的偏差である．実験に使用した計算環境は，OS: Ubuntu 18.04 (64bit), CPU: Intel Core i7-6700 (3.4GHz), メモリ: 8GB である．

4.　おわりに

線形順序付け問題に対する分解定理 (定理 1.1) は, 入力行列の分解によって計算時間の短縮が可能であることを示唆しているが, 分解定理による入力行列のサイズの縮小に失敗する問題例が多く存在する．本研究ではそのような問題例に対して, 分解定理で与えられる分解と類似する分解によって問題サイズの縮小が可能となる発見的解法: レベルグラフを用いるもの, 及び, 最小カットに基づくものを提案した．予備実験の結果はこれらの発見的解法の著しい有効性は認められなかったが, さらなる性能の改善が可能であると思われる．

謝辞

本研究は JSPS 科研費 18K11180 の助成を受けたものである．

参考文献

[1] E. Garcia, J. Ceberio and J.A. Lozano: Hybrid heuristics for the linear ordering problem. *2019 IEEE Congress on Evolutionary Computation* (2019), pp. 1431–1438.

[2] K. Hanauer: *Linear Orderings of Sparse Graphs.* Dissertation, Fakultät für Informatik und Mathematik, Universität Passau, 2018.

[3] R. Martí and G. Reinelt: *The Linear Ordering Problem: Exact and Heuristic Methods in Combinatorial Optimization* (Springer, 2011).