線形配置問題に対する改良型ヒューリスティック
Improved Heuristic for Linear Arrangement Problem

田中 僚 * 山口 一章 * 増田 澄男 *
Ryo Tanaka Kazuaki Yamaguchi Sumio Masuda

1　はじめに

無向グラフ $G = (V, E)$ の各辺 $(v_i, v_j) \in E$ に重み $w(u_i, v_j)$ が与えられているとする．V の要素を一列に並べ，順に $v_1, v_2, \ldots, v_n$ とするとき，辺 (v_i, v_j) の加重辺長を $w(v_i, v_j) \cdot |i - j|$ と定める．また，全ての辺に対するこの値の和を，並びの加重辺長総和と呼ぶことにする．各辺が重みを持つ無向グラフが与えられたときに加重辺長総和が最小となるような頂点の並びを求める問題は線形配置問題（Linear Arrangement Problem，以降 LAP と略記する）と呼ばれる．以下，図 1 に入力例（左）と最適解（右）を示す．辺重みは全て 1 としている．図左の並び順をそのまま頂点の並びとしたときは加重辺長総和は 5 であるが，図右の並び順においては加重辺長総和は 3 であり，この入力に対する最小値である．以降，加重辺長総和をコストと書く．

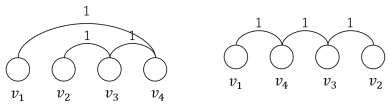

図 1　入力例と最適解

一般のグラフに対する LAP は NP 困難であることが知られている [4]．LAP の応用としては，施設配置，生物学的応用，グラフ描画，ソフトウェアダイアグラムレイアウト，ジョブスケジューリングなどが知られている [1][2][3]．

本稿では LAP に対し，局所探索法をベースとした改良型ヒューリスティックの一つであるランダム局所探索法について，いくつかの近傍を用いて LAP を解いたときの振る舞いについて，計算機実験によって比較する．

2　局所探索法

局所探索法は最適化問題に対する発見的手法の枠組みの一つであり，各実行可能解に対して何らかの方法で近傍と呼ばれる解集合を定める．局所探索法は，はじめにある実行可能解を何らかの方法で一つ生成し暫定解とし，その近傍の中により良い解があればそれを新たな暫定解とするという処理を繰り返すことによってより良い解を目指すアルゴリズムである．近傍の定め方によっては，最適解でないにも関わらず，近傍により良い解が存在しないような解にたどり着くことがある．このような解は局所最適解（または局所解）と呼ばれる．

局所探索法を最適化問題に適用する際，問題毎に経験的に良いとされている近傍が示されている．LAP のような順列を求める組み合わせ最適化問題に対する近傍としては，交換近傍や挿入近傍が知られている．

挿入近傍は，ある頂点を別の場所に挿入し，以降の頂点を後ろにずらして得られる解からなる近傍である．頂点の選択と挿入場所の組合せは n^2 通りあるが，i 番目の頂点を一つ後ろにずらすのと $i+1$ 番目の頂点を一つ前にずらすのは同じ解を与えるため，重複を除いた組合せの数は $n^2 - n + 1$ 通りである．交換近傍は，二つの頂点を入れ替えてできる解からなる近傍であり，組み合わせは n 個の頂点から二つの頂点を選ぶ $n(n-1)/2$ 通りである．本稿では挿入近傍により新たな解を得る操作を $Insert(i, j)$，交換近傍が得られる操作を $Swap(i, j)$ と表記する．

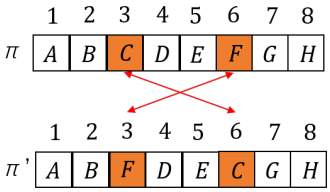

図 2　解 π と $Swap(3, 6)$ 実行後の解 π'

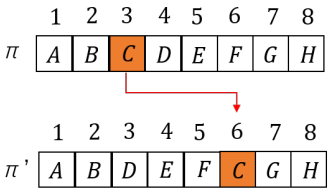

図 3　解 π と $Insert(3, 6)$ 実行後の解 π'

3　ランダム局所探索

前述の通り，組合せ最適化問題に対して局所探索法を用いた場合，近傍の定め方によっては局所解に陥ってそれ以上解の更新が行われなくなることがある．それを回避する手段としては以下などが知られている．

1. 解を悪くするような遷移を許す（タブー探索，模擬焼きなまし法, ランダム局所探索法など [8]）

* 神戸大学大学院工学研究科 Graduate School of Engineering, Kobe University,Kobe-shi 657-8501,Japan

2. 近傍を広げるなどして局所解よりよい解が見つかるようにする（可変近傍探索法）
3. 目的関数を本来のものと少し異なるものにして局所解から脱出させる（誘導局所探索法）.
4. 全く異なる初期解を作りそこから探索を開始する方法（多スタート法）.

本稿では，ランダム局所探索法を扱う．ランダム局所探索法は乱数によって定まる小さな近傍の中の最良解を調べる．その際，現時点の暫定解より良い解であればそれを新たな暫定解とする．現時点より悪い場合でも，確率的に新しい解を暫定解として採用する．評価値の悪い解を採用する確率を徐々に下げることによって良い解に収束することが期待される．おおよその振る舞いは模擬焼きなまし法と同じであり，変種の一つといえる．違いは，近傍を極めて小さなものに制限するところにある．この方法は線形順序付け問題に対して有効であることが示されている [5].

近傍のうち目的関数が最適値に近づく解候補のみを選択する山登り法とは違い，最適解から遠ざかる近傍にも低確率で遷移を許すことで，探索空間の多様性を向上させる．

LAP に対する，挿入近傍によるランダム局所探索法のアルゴリズムは以下の一例を以下に示す．まず，初期順列をランダムに生成し，その初期解を π とする．そして許される限り，以下のステップを繰り返す．

1. 1 以上 n 以下のランダムな整数値 i を生成し，移動候補の頂点を $\pi(i)$ とする．
2. $\pi(i)$ を，現在の位置以外の全ての位置 j に移動したときのコストを計算し，コストが最小になる位置 j_{best} とコストの増分 $\Delta cost$ を記憶.
3. $\Delta cost$ が負なら，$Insert(i, j_{best})$ を実行.
4. $\Delta cost$ が正なら，ある確率 p で $Insert(i, j_{best})$ を実行.

上記 1. ∼ 4. の時間計算量を評価する．ステップ 1 は $O(1)$ 時間で実行可能である．ステップ 2 については $O(n)$ 時間である（詳細は 6.1 で述べる）．ステップ 3，4 では移動距離を d とすると，$O(d)$ である．したがって全体の時間計算量は $O(n)$ となる．通常の挿入近傍や交換近傍は $O(n^2)$ 個の解を調べるが，それらに比べて時間計算量が小さくて済む.

挿入近傍において狭い近傍を用いることは，局所解に陥ることを防ぐ効果もある．挿入近傍において評価値が悪くなるようなを移動を行うとき，その直後に同じ頂点が選ばれると解が元に戻ってしまう．しかしそれが起きる確率は $1/n$ であり，入力のグラフがある程度大きければ元に戻る確率は十分小さくなる.

4 移動の制限

文献 [5] では LAP に対するランダム局所探索において移動の制限を加える方法について考察されている．以下でそれらを紹介する.

前述のランダム局所探索法において，挿入する頂点 i と挿入される位置 j の距離 $|i-j|$ が小さいと，解の遷移が巡回しやすくなる傾向がある．改悪する際に頂点の移動距離を，ある程度以上の場合だけに制限することによりそれを回避できる．移動距離の下限 d_{min} を $0.1n \leq d_{min} \leq 0.5n$ を満たすように設定すると良好な結果が得られることが実験により確認されている．この制限を加えた方法を手法 1 とする．また，頂点の移動距離を $0.5n$ で固定する方法も提案されている．この手法を手法 2 とする.

以下に，LAP に対するランダム局所探索法のアルゴリズムの疑似コードを記載する.

Algorithm 1 LAP に対するランダム局所探索法

```
π ← ランダムに生成した初期解
cost ← πのコスト計算結果
π_best ← π
cost_best ← cost
while !終了条件 do
  i ← 移動する頂点の元の位置 i を 1 以上n以下の整数からランダムに決定
  コスト最小の挿入先を j，挿入時の評価値の増分を Δc とする.
  if Δc < 0 || 確率 p の抽選 then
    π ← Insert(i, j)
    cost ← cost + Δc
    if cost < cost_best then
      π_best ← π
      cost_best ← cost
    end if
  end if
end while
return π_best
```

5 提案法

本稿では LAP に対するランダム局所探索法において，解に摂動を加える方法を提案する．確率 q である長さの頂点順列を逆順にすることで，解に突然変異を与えるヒューリスティックを提案する．決まった範囲を逆順にする理由として，ランダム局所探索法よりも更にもとの解に戻りにくくすることによって，ランダム局所探索以上に解に多様性が生まれ，広い解空間を探索することができると考えたからである.

逆順にする確率 q の決定は実験の結果で行った．あるベンチマーク問題に対して $q = 1, 0.1, 0.01, 0.001, 0.0001$ の 5 通りで 10 回実験し，出力の平均コストを比べ，最も良い結果が出た $q = 0.1$ を採用した.

また，逆順にする範囲だが，入力グラフのサイズが入力ごとに異なるため，固定長を採用することはできない．そこで，本研究では入力グラフの頂点数を n とすると，$n/4$ から $3n/4$ の範囲の頂点を逆順にする方法を採用した．中央付近の頂点群は端に配置されている頂点よりも繋がっている辺が多く，中央の $n/2$ 個の頂点を逆順にすることで評価値の変動を大きくすることができ，解空間を広く探索できると考えたためである.

次ページに提案法のアルゴリズムの疑似コードを記載する.

Algorithm 2 LAP に対する提案法

$\pi \leftarrow$ ランダムに生成した初期解
$cost \leftarrow \pi$のコスト計算結果
$\pi_{best} \leftarrow \pi$
$cost_{best} \leftarrow cost$
while !終了条件 **do**
  if 確率 q の抽選 **then**
    $(\pi, \Delta c) \leftarrow$ 一部を逆順にした場合の順列とその際のコスト変化Δc
    $cost \leftarrow cost + \Delta c$
  end if
  $i \leftarrow$ 移動する頂点の元の位置 i を $[1, N]$ の整数からランダムに決定
  $(j, \Delta c) \leftarrow$ コスト最小の挿入先 j と挿入時のコスト変化Δc
  if $\Delta c < 0 ||$ 確率 p の抽選 **then**
    $\pi \leftarrow Insert(i, j)$
    $cost \leftarrow cost + \Delta c$
    if $cost < cost_{best}$ **then**
      $\pi_{best} \leftarrow \pi$
      $cost_{best} \leftarrow cost$
    end if
  end if
end while
$return \pi_{best}$

以下に頂点順列を逆順にした際の具体例を挙げる．

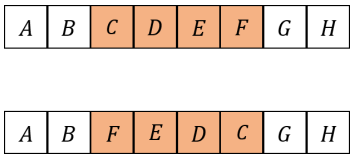

図 4　頂点順列の反転例

5.1　近傍計算

今回のアルゴリズムでは，挿入される頂点 i はランダムに一つ決まるが，挿入先の位置 j は，評価値が最も良いとこに決まる．遷移先の評価値を高速に計算しなければ，探索のループ回数やグラフの頂点数が増えるにつれ膨大な計算時間が必要となってしまう．

そこで，遷移先の評価値の高速な計算方法について，以下を提案する．

1. それぞれの頂点 v に順列の先頭側に伸びる辺の数 $LE(v)$ と順列の末尾側に伸びる辺の数 $RE(v)$ を記憶させる．
2. 頂点 i と頂点 $i+1$ の間に辺が存在すれば，$RE(i) = RE(i) - 1$，$LE(i+1) = LE(i+1) - 1$ を実行
3. $\Delta c = LE(i) - RE(i) + RE(i+1) - LE(i+1)$
4. 2 を実行したなら，$RE(i+1) = RE(i+1) + 1$，$LE(i) = LE(i) + 1$ を実行

$Insert(x, y)$ を実現させるには，上記の操作を $i : x \rightarrow y - 1$ まで繰り返せばよい．例えば，$Insert(2, 5)$ を実行する場合は $i : 2 \rightarrow 4$ の間繰り返す．

6　計算機実験

6.1　実験環境

ベンチマーク問題として，DIMACS のクリークのグラフ [7] を用いて実験，比較を行った．辺重みは 1 である．実験環境は以下の通りである．

CPU : A10-7800 3.5GHz，メモリ : 8.0GB，OS : Windows10，使用言語 : Java8.0.

改悪を許す実行確率は $p = 0.01$，$d_{min} = 0.5n$ とした．

また，逆順を行う確率は $q = 0.1$ とした．

6.2　実験結果

RLS : ランダム局所探索法

手法 1 : $d_{min} = n/2$

手法 2 : 移動距離 $n/2$ で固定

表 1 は，ステップ数 10^6 回のもとで実験した際のそれぞれの手法におけるコストを比較したものである．

表 2 は提案法を 10^6 ステップ回した際にかかった実行時間を示したもので，単位は秒 (s) である．

表 1　各方法におけるコストの比較

Instance	頂点数	RLS	手法 1	手法 2	提案法
C125.9.clq	125	280062.1	279410.6	279582.6	279306.3
C250.9.clq	250	2271643.3	2268388.1	2269311.6	2266769.2
C500.9.clq	500	18360108.3	18353998.0	18355937.2	18343050.1
Mann_a27	378	8860101.0	8856729.0	8856095.6	8854092.4
block200_2	200	596375.4	595021.7	594824.7	592994.0
block200_4	200	815654.4	814499.7	814210.8	813037.8
keller4	171	483559.2	483976.6	483902.8	483397.2
keller5	776	54423012.6	54461912.8	54464802.0	54460289.1
p_hat300-1	300	859473.9	858202.7	858239.0	857929.3
p_hat300-2	300	1828501.4	1824152.3	1825779.6	1823572.0

表 2　各インスタンスにおける提案法の実行時間 (s)

Instance	頂点数	提案法
C125.9.clq	125	678.2
C250.9.clq	250	1268.6
C500.9.clq	500	1786.7
Mann_a27	378	1569.4
block200_2	200	1079.5
block200_4	200	1023.0
keller4	171	892.1
keller5	776	2031.5
p_hat300-1	300	1469.1
p_hat300-2	300	1397.3

6.3　考察

表 1 から，ランダム局所探索法，手法 1，手法 2 と同じステップ数で実験すると，突然変異を与えた提案法では，すべてのインスタンスに対してより小さいコストを求めることができたことがわかる．ランダム局所探索法や手法 1，2 では，確率 $p = 0.01$ でのみ改悪を許すため，広い解空間を探索できるものの、その探索に多くのステップ数がかかってしまう．一方，提案法では逆順処理という突然変異を加えるため，解が大幅に変

わり，提案法では既存の手法で発見できなかったような局所解を見つけることに成功していると考えられる．

また，サイズの小さなインスタンスについて，ランダム局所探索法を 10^8 ステップ回した実験結果よりも提案手法を 10^6 ステップ回したの実験結果のほうが良い結果が出ている．このことから，突然変異を加えることによって少ないステップ数で良い解をより効率よく求められることが分かった．

7　まとめと今後の課題

既存のランダム局所探索法ベースのヒューリスティックと提案法を同じ評価値を用いて実験し，その結果を比較した．使用したベンチマーク問題について同じステップ数で実験すると，すべてのインスタンスで提案法のほうが既存の手法よりも良い結果を出力した．これはある長さの頂点順列を逆順列にすることで，既存の手法よりも効率よく広い範囲の解空間を探索できたからだと考えられる．

今後の課題として，高い解の精度を保ちながら 10^6 よりも少ないステップ数で解が収束するようなアルゴリズムの考案が挙げられる．現在，それぞれの頂点に順列の先頭側に伸びる辺の数 LE を記憶させているため，$LE(v)$ の大きい頂点 v から逆順を行うことにより効率よく探索が行われると考えられる．逆順を行う範囲の外側との関係が密である頂点を選択することで，より評価値を悪化させることができ，少ないステップ数で広い解空間を探索できると考えるからである．今後はこの方法の実装と，さらなる高速化，さらに質の良い解を求める方法を考えていきたいと思う．

参考文献

[1] 本間 裕大，施設配置の数理，Operations research as a management science research 57(6),327-334,2015-09-01

[2] Y.-L. Lai, K. Williams, “A survey of solved problems and applications on bandwidth edgesum and profile of graphs Graph Theory,” 31 (1999), pp. 75-94

[3] J. Diaz, J. Petit, M. Serna, “A survey of graph layout problems,” ACM Computing Surveys, 34 (3) (2002), pp. 313-356

[4] M.R. Garey, D.S. Johnson, Computers and intractability: a guide to the theory of NP-completeness W.H. Freeman, New York (1979)

[5] 兼本樹，線形順序付け問題と線形配置問題に対する局所探索法の近傍操作の提案，平成 31 年度神戸大学大学院工学研究科修士論文．2019

[6] 兼本　樹, 山口　一章, 増田　澄男 線形順序付け問題に対する焼きなまし法の近傍選択法,’ 平成 30 年電気関係学会関西連合大会, G10–4, 2018.

[7] DIMACS BenchMark Set, http://iridia.ulb.ac.be/f̃mascia/maximum_clique/DIMACS-benchmark, 2019/7 アクセス

[8] 久保 幹雄，J.P. ペドロソ，メタヒューリスティックの数理，共立出版 (2009)