

A-029

ノード配置問題に対する交叉法とパス再結合法の性能比較

Performance Comparison of Crossover and Path-Relinking for the Node Placement Problem

赤木 勇斗†
Yuto Akagi

片山 謙吾‡
Kengo Katayama

南原 英生†‡
Hideo Minamihara

西原 典孝†‡
Noritaka Nishihara

1. まえがき

近年、ネットワークの高速かつ大容量の通信を実現する技術として波長分割多重 (Wavelength Division Multiplexing, WDM) 技術が注目されている。この WDM の論理トポロジーとして双方向マンハッタンストリートネットワーク (Bidirectional Manhattan Street Network, BMSN) がある。我々の研究グループでは BMSN を対象としたノード配置問題 (Node Placement Problem, NPP) に対して k -opt 局所探索法 (k -opt Local Search, KLS) と遺伝的アルゴリズム (Genetic Algorithm, GA) をハイブリッドした Memetic アルゴリズム (Memetic Algorithm, MA) を提案し、高い探索性能が得られることを示している [1]。この MA に含まれる操作の一つに、2つの解を組み合わせることで新たな解を生成する交叉法 (Crossover) がある。このように複数の解を組み合わせ新たな解を生成する代表的な操作法には、交叉法の他にパス再結合法 (Path-Relinking, PR) が知られている。これらの方法は解の構造や設計者のアイデア等に委ねられ、自由度が高く様々な方法が考えられるため、より綿密な検討が重要である。また、GA 等の枠組みのもとで交叉法とパス再結合法について比較検討した報告例も少ない。そこで本研究では、NPP に対する MA の枠組みのもとで文献 [1] で提案した交叉法と新たに提案するパス再結合法の性能比較を行う。

2. BMSN のノード配置問題 NPP

ネットワークに関連する組合せ最適化問題の一つとして、双方向マンハッタンストリートネットワーク (BMSN) を対象とするノード配置問題 (NPP) があり、NP-困難であることが知られている。BMSN は、 $X \times Y$ ($= n$) 個のネットワークノードが規則的に格子状に配置された論理トポロジーを持ち、各ノードは隣接する 4 ノードと双方向に接続され、端のないトーラス構造をなしている (図 1 参照)。本論文では、 $X = Y$ ($= m$) の BMSN グラフを対象とする。BMSN の NPP は、ノード i からノード j ($i \neq j$) へのトラヒック量を表す $n \times n$ 行列 $T = [t_{i,j}]$ ($0 \leq i, j \leq n-1$) が与えられた時、式 (1) の目的関数を最小化するノード配置 σ を求める問題である。

$$f(\sigma) = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} t_{i,j} \times h(\sigma_i, \sigma_j) \quad (1)$$

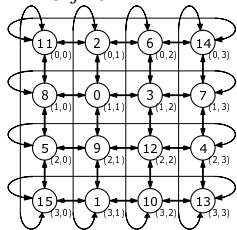


図 1 BMSN グラフの一例

ただし、 σ は n 要素 $\{0, 1, \dots, n-1\}$ からなる順列で、 $\sigma_i = j$ はノード i が BMSN グラフ上の j 番 (座標 $(\lfloor j/m \rfloor, j \bmod m)$) の頂点 (または場所、スロットとよぶ) に配置されることを表す。図 1 の場合、 $\sigma = \{5, 13, 1, 6, 11, 8, 2, 7, 4, 9, 14, 0, 10, 15, 3, 12\}$ となる。 $h(\sigma_i, \sigma_j)$ は σ_i と σ_j に配置されたノード i, j 間の最短経路ホップ数を表す。

†岡山理科大学大学院 工学研究科
‡岡山理科大学 工学部 情報工学科

3. NPP に対する交叉法

本論文で比較対象とする交叉法 [1] を説明する。本論文では、交叉法に対して与えられた 2つの解を“親”、生成される 2つの解を“子”と表現する。この交叉法の基本方針は、NPP の解 σ が表す図 1 のようなネットワークを無向グラフ $G_\sigma = (V, E)$ と捉えた上で、2つの親に共通して存在するノードの隣接関係を子に引き継がせることである。具体的な手順を以下に示す。

まず、親 σ^a と σ^b に共通して存在するノードの隣接関係を表す枝、即ち“共通リンク”を探す。図 2 は親 σ^a と σ^b の共通リンクを太線 (赤) で示している。

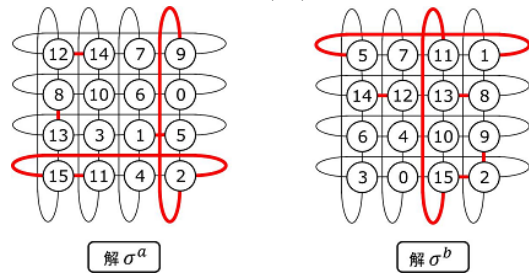


図 2 親 σ^a と σ^b の共通リンク

この図 2 の共通リンクを持つノード $\{1, 2, 5, 8, 9, 11, 12, 13, 14, 15\}$ の配置を、2つの子 σ^c と σ^d にそのまま引き継ぐ。次いで、図 3 のように残りのノード $\{0, 3, 4, 6, 7, 10\}$ をランダムに未割当の場所へ配置することで、子の生成を終了する。

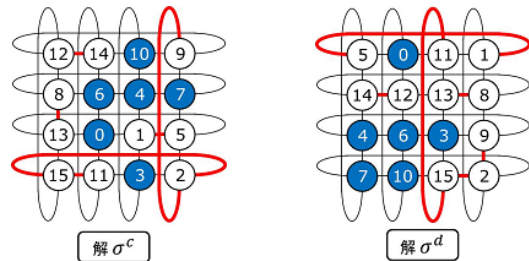


図 3 交叉法により得られた 2つの子 σ^c と σ^d

4. NPP に対するパス再結合法

パス再結合法 (PR) と交叉法は複数の解から新たな解を生成するという点において共通しているが、解の生成法は異なる。まず、その PR の一般的な解の生成法を示す (図 4)。

本論文では、PR に対して与えられた 2つの解をそれぞれ“スタート解 σ^s ”と“ターゲット解 σ^t ”、生成される 1つの解を“ σ^p ”と表現する。スタート解 σ^s からターゲット解 σ^t へと解を変換する際の最短距離 (最小近傍操作回数) を l とした時、変換過程で得られる解集合を $\sigma^1, \sigma^2, \dots, \sigma^{l-1}$ とする。この解集合 $\sigma^1, \sigma^2, \dots, \sigma^{l-1}$ をスタート解 σ^s からターゲット解 σ^t への“パス”と呼び、このパス上の解が PR で得られる解 σ^p の候補となる。このように交叉法が複数の解の共通部分をもとにして新たな解を生成するのに対して、PR は生成されるパス上の解を PR で得られる新たな解の候補とする。

本論文の NPP に対する PR では、3. 節の共通リンク

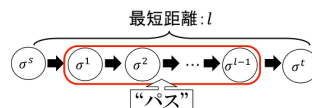


図 4 パス再結合法の概略図

に注目した交叉法とは異なり、ノードの配置場所に注目する。NPP に対する PR の手順を図 5 に示す。この PR ではスタート解 σ^s とターゲット解 σ^t で配置の異なるノード同士の交換操作を、スタート解 σ^s 上で繰り返すことにより、 σ^s から σ^t へのパスを生成する。図 6 では σ^s と σ^t で配置の異なるノード $\{0, 3, 8, 9, 10, 12, 15\}$ を色分けしている (赤)。図 6 の色分けされたノードは、図 5 の Line 3 の“選択候補ノード集合 P_{node} ”に格納される。この P_{node} に含まれるノードがパス生成時に交換操作の対象となるが、生成されるパスは交換するノードの順序に大きく依存する。そのため、本論文ではパスの生成過程において“パス生成戦略”というルールを設けており、このルールにもとづきノードを P_{node} から交換操作毎に選択する (Line 4 ~ 8)。本論文では、このパス生成戦略の違いからなる 2 種類の PR (ランダム PR, 最良移動戦略 PR) を提案する。それらの詳細は後述する。次いで、生成されたパスの中から“解選択法”にもとづき解を 1 つ選択し、PR で得られる解 σ^p とする (Line 9)。この解選択法の例としては、最良解、最悪解やランダム解を選択する方法が挙げられる。

```

procedure Path-relinking( $\sigma^t, \sigma^s$ )
begin
1  選択候補ノード集合  $P_{node}$  を空集合として初期化する
2  解集合 (パス)  $P_{path}$  を空集合として初期化する
3  スタート解  $\sigma^s$  とターゲット解  $\sigma^t$  で配置の異なるノード を集合  $P_{node}$  に加える
4  repeat
5    ノード集合  $P_{node}$  から “パス生成戦略”にもとづきノード  $i$  を選択する
6    ノード  $i$  を用いてパスを生成する交換操作を行い、解  $\sigma^r (1 \leq r \leq l)$  を得る
7    解  $\sigma^r (r \neq l)$  を解集合  $P_{path}$  に加える
8  until  $\sigma^t = \sigma^r (r = l)$ 
9  解集合  $P_{path}$  から “解選択法”にもとづき解を選択し  $\sigma^p$  とする
10  $\sigma^p$  を返す
end
    
```

図 5 NPP に対する PR の疑似コード

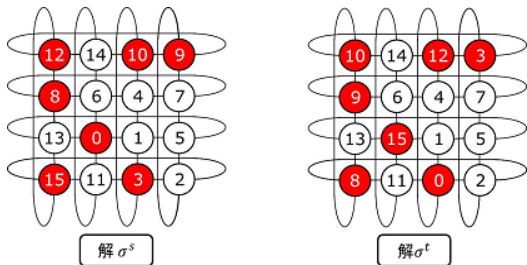


図 6 選択候補ノード

以下に本論文で提案するパス生成戦略が異なる 2 種類の PR (ランダム PR, 最良移動戦略 PR) を示す。

ランダム PR: 選択候補ノードから、ランダムにノードを選択する。

最良移動戦略 PR: 選択候補ノードから、交換後に行われる解の中で評価値が最も良い解に対応したノードを選択する。

なお、本論文では、上述の PR によって解 σ^p を得た後に、ターゲット解 σ^t からスタート解 σ^s へのパスを生成する PR を行うことで 2 つ目の解 σ^p を生成する。よって、この PR で得られる解は計 2 つとなる。

5. 実験結果

NPP に対する交叉法と PR の性能を比較検討するために、実験を行った。実験は文献 [1] の MA の枠組みに、本論文で示した交叉法と 2 種類の PR をそれぞれ導入して行う。本論文では各 PR の解選択法を最良解、最悪解としてそれぞれ実験を行った。使用問題例はベンチマーク問題例 [2] から、16, 64, 256 の問題サイズをそれぞれ 20 例題とした。本実験では各問題に対して、10 回ずつ試行した。MA の個体数は 10、突然変

異処理は文献 [3] の CrossKick with Greedy を用いており、最良解の更新が 10 世代以上行われなければ多様化/再スタート戦略 [1] を行う。局所探索法は文献 [4] の枝刈り法 3 を導入した変形 KLS を使用する。本実験ではこの KLS に、 k -opt 近傍探索中に \sqrt{m} (問題サイズ $n = m \times m$) 回以上連続で改善解が得られなかったとき、 k -opt 近傍探索を打ち切るという枝刈りを追加した。MA の終了条件は計算打切時間を設定し、各問題サイズ (16, 64, 256) それぞれに対して 1 秒, 10 秒, 100 秒とした。使用計算機は HP Workstation Z420 (CPU: Intel Xeon E5-1620 3.6GHz, RAM: 4GB) であり、アルゴリズムは C 言語によりコード化した。

表 1 に実験結果を示す。各表の欄 operator は交叉法または PR (括弧内は解選択法)、 n は問題サイズ、 opt は各問題サイズの最適解の評価値、 $best$ は 20 例題をそれぞれ 10 回試行して得られた最良解の評価値、 avg は 20 例題をそれぞれ 10 回試行して得られた解の平均評価値、 $gen.avg$ は 1 回試行に要する MA の平均実行世代数である。表 1 から $best$ において問題サイズ 16・64 では全ての解法で最適解を算出した。問題サイズ 256 では交叉法とランダム PR (最悪解) が算出した値 (315) が最も良好であった。 avg で比較すると、問題サイズ 16 では全ての解法で最適解を算出しており、問題サイズ 64 では最良 PR (最悪解)、問題サイズ 256 ではランダム PR (最悪解) がそれぞれ最良の値を算出した。また、各問題サイズそれぞれの $gen.avg$ の値は、各解法間で大きな差はないことから、各解法の処理速度はほぼ同程度であったことが分かる。

以上の結果から、問題サイズ 64・256 において良好な解を算出したのはいずれも PR であったことを示した。また、PR のみの結果に着目すると、解選択法を最悪解にした各 PR が比較的良好な解を算出した。特にランダム PR (最悪解) は本実験で最大の問題サイズ 256 の $best$, avg において最も良好な値を算出している。これらのことから PR が交叉法と同等以上の解を算出できることを示した。

表 1 実験結果

operator	n	opt	$best$	avg	$gen.avg$
交叉法	16	19	19	19.0	1306.5
	64	76	76	77.2	970.4
	256	307	315	340.0	464.5
ランダム PR (最良解)	16	19	19	19.0	1297.5
	64	76	76	77.1	967.8
	256	307	320	340.2	470.7
ランダム PR (最悪解)	16	19	19	19.0	1279.4
	64	76	76	77.1	934.2
	256	307	315	332.3	406.6
最良 PR (最良解)	16	19	19	19.0	1308.6
	64	76	76	77.2	962.1
	256	307	319	341.4	444.5
最良 PR (最悪解)	16	19	19	19.0	1291.6
	64	76	76	76.9	952.2
	256	307	318	336.8	428.6

6. むすび

本論文では、NPP に対する MA の枠組みのもとで交叉法 [1] と新たに提案した 2 種類の PR の性能比較を行った。比較実験の結果から、PR が交叉法と同等以上の解を算出できることを示した。今後の課題として PR の新たなパス生成戦略や解選択法の検討、他の組合せ最適化問題への応用等が挙げられる。

参考文献

- [1] 山下浩司, 片山謙吾, 南原英生, 成久洋之, “ノード配置問題に対する Memetic アルゴリズム,” 進化計算シンポジウム 2007 講演論文集, pp. 99-102, 2007.
- [2] NPP のベンチマーク問題例集, <http://k2x.ice.ous.ac.jp/~katayama/bench/npp/>
- [3] 大倉慶一, 片山謙吾, 南原英生, 西原典孝, “ノード配置問題に対する突然変異の検討,” 平成 23 年度電気・情報関連学会中国支部第 62 回連合大会講演論文集, pp.158, 広島, 2011.
- [4] 赤木勇斗, 渡邊好幸, 片山謙吾, 南原英生, 西原典孝, “ノード配置問題に対する k -opt 局所探索法の枝刈りによる効率化,” 平成 24 年度電気・情報関連学会中国支部連合大会講演論文集, pp.259-260, 松江, 2012.