

最適化アルゴリズムによる 低遅延相互結合網のためのグラフ構成法

中尾 昌広^{1,a)} 酒井 真章² 花田 良子² 村井 均¹ 佐藤 三久¹

概要: 大規模並列計算機システムやいくつかの工業製品内における様々なネットワークにおいて、直径と平均ホップ数が小さいネットワークを適用することにより、システム全体の性能が向上することが期待できる。ネットワークは無向グラフとしてモデル化することで、グラフ理論上の Order/Degree 問題として定義できる。本稿では、Simulated Annealing をベースにした Order/Degree 問題に対する最適化アルゴリズムを提案する。本アルゴリズムの特徴は、ネットワークに対称性を持たせることにより、最適化アルゴリズムの探索性能を高めている点である。Order/Degree 問題のための国際コンペティション Graph Golf が出題しているグラフを用いて本アルゴリズムの評価を行った。その結果、本アルゴリズムは直径と平均ホップ数が十分に小さいネットワークを生成できることを示した。さらに、生成されたネットワークに対して、シミュレーションによるレイテンシと並列ベンチマークの性能評価を行った。その結果、ランダムネットワークと比較して、最適化されたネットワークの方が高い性能を発揮することを示した。

1. 背景

スーパーコンピュータやデータセンタなどの並列計算機システムの大規模化により、計算ノード間を接続する相互結合網のレイテンシが並列アプリケーションの性能に強く影響を与えている。近年の研究では、ランダムトポロジが持つスモールワールド性を、並列計算機システムを含めた様々な工業製品のネットワークに応用する研究が行われている [1-4]。並列計算機システムの場合、ランダムトポロジを採用することにより、相互結合網の直径と平均ホップ数 (ASPL: Average Shortest Path Length) が従来の k -ary n -cube などの規則的なトポロジよりも小さくなるため、並列アプリケーションの性能が向上することが報告されている。ただし、ほとんどの場合、ランダムトポロジの直径と ASPL はそれらの理論的な下界よりも大きな値をとるため [5]、直径と ASPL がより小さいグラフを設計することが重要である。また、物理的な制限 (例えば、InfiniBand-EDR のメタルケーブルの最大長は 3m である) やハードウェア量と消費電力を抑えるため、それらのネットワークケーブルの最大長を考慮する場合がある [6, 7]。

そのような背景のもと、グラフ理論の問題の 1 つとして

Order/Degree 問題が提案されている [8]。Order/Degree 問題とは、与えられた頂点数 (Order) と次数 (Degree) を満たす無向グラフの集合の中から、最小の直径と ASPL を持つグラフを発見する問題である。例えば、並列計算機システムの計算ノードを「頂点」、ネットワークのケーブルを「エッジ」とみなすことで、そのネットワークトポロジは無向グラフとして表現できる。このことから、Order/Degree 問題から発見されたグラフは様々な工業製品への応用が期待できる。Order/Degree 問題のカテゴリには、頂点を自由に配置できる一般グラフと、頂点を 2 次元格子の節点上に配置する格子グラフがある。一般グラフではすべての頂点間の距離は等しいのに対し、格子グラフでは頂点間の距離はマンハッタン距離で定義される。マンハッタン距離である理由は、斜め配線が許されない並列計算機システムやチップ内ネットワークを想定しているためである。また、格子グラフではエッジの最大長が設定可能であるため、ネットワークケーブルなどの最大長を考慮した設計を行うことができる。

Order/Degree 問題の定義は簡易であるが、次の理由により、その効率的な解法は発見されていない。(1) 与えられた頂点数と次数を満たすグラフの数は膨大であるため、すべての組合せを調べることはできない。(2) Order/Degree 問題を離散最適化問題として定義した場合、多数の局所解が存在するようなランドスケープを持つため、大域的最適解を発見することが難しい。(3) 直径と ASPL を求めるた

¹ 理化学研究所 計算科学研究センター

RIKEN Center for Computational Science

² 関西大学 システム理工学部 電気電子情報工学科

Kansai University, Faculty of Engineering Science, Department of Electrical, Electronic and Information Engineering

a) masahiro.nakao@riken.jp

めの計算量が大きい。全頂点間のホップ数を計算することで直径と ASPL を求めることができるが、その計算量は頂点数の 2 乗に比例するため、膨大な計算時間が必要になる。

本研究の貢献は、直径と ASPL がランダムトポロジと比べて十分に小さく、かつ任意の頂点数と次数を持つ一般グラフおよび格子グラフを生成できるアルゴリズムを提案することである。本アルゴリズムの特徴は次の通りである。

(1) 汎用的な最適化アルゴリズムの一種である Simulated Annealing (SA) [9,10] を用いることで、局所解に容易に陥らない解探索を行う。(2) グラフに対して対称性を与えることで、SA の解探索能力を高め、かつ直径と ASPL を求めるための計算量を大幅に削減する。

我々の過去の研究 [11] では、一般グラフを対象とした手法を提案している。過去の研究と本研究との差分は次の通りである。(1) 過去の研究で提案した手法を格子グラフも扱えるように拡張する (3.1 節)。(2) 過去の研究では、素数の頂点数を持つ一般グラフについては対称性を用いた解探索を行うことができなかった。本研究では、そのようなグラフも扱えるように拡張する (3.3 節)。(3) 過去の研究において、評価に用いたグラフは最大頂点数が 256 という小規模グラフであった。本研究では、頂点数が最大 100 万の大規模グラフを用いた評価を行う (4.1 節)。(4) 並列シミュレーションフレームワーク SimGrid [12] を用いて、生成されたネットワークにおける並列アプリケーションの性能について評価する (5 章)。

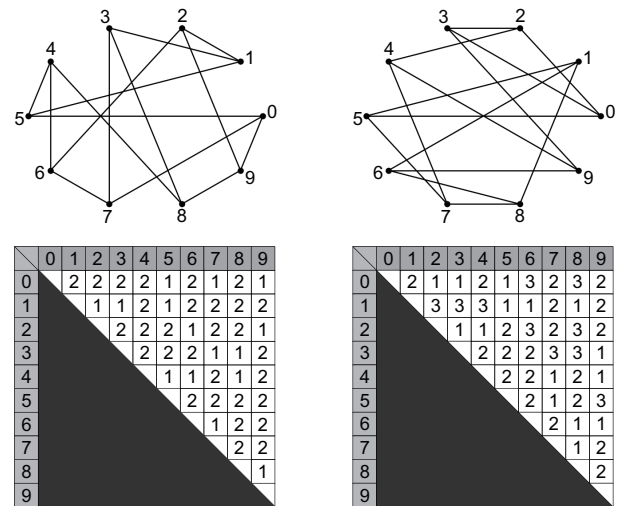
本稿の構成は次の通りである。2 章では Order/Degree 問題の概要とその関連研究について述べる。3 章では提案手法について述べる。4 章では提案手法の解探索性能の評価を行う。5 章では生成されたネットワークに対する評価を行う。6 章では本稿のまとめと今後の課題について述べる。

2. Order/Degree 問題

2.1 概要

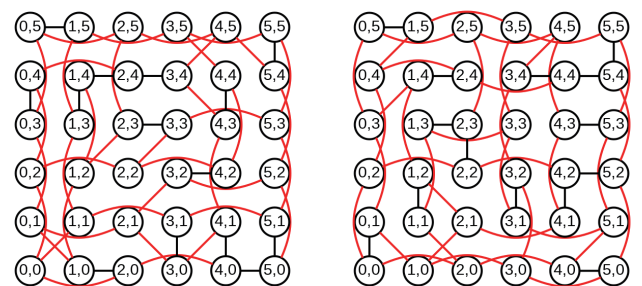
Order/Degree 問題とは、指定された頂点数 (n) と次数 (d) を満たすグラフの中から、最小の直径を持つグラフを選択する問題のことである。なお、同じ直径を持つグラフが複数ある場合は、最小の ASPL を持つグラフを選択する。格子グラフの場合、各頂点は 2 次元格子の節点上に配置されるため、2 次元格子の横 (w) と縦 (h) には、 $n = w \times h$ という関係が成り立つ。さらに、頂点間を結ぶエッジにはマンハッタン距離で定義される最大長 (r) も設定できる。

一般グラフ $(n, d) = (10, 3)$ の例とその距離行列を図 1 に示す。距離行列とは、頂点間の最短ホップ数を示した表である。Order/Degree 問題では無向グラフを扱うため、その距離行列は対称行列になる。この距離行列の要素の最大値が直径であり、全要素の合計値を要素数 $(n^2 - n)/2$ で割った値が ASPL である。図 1 においては、より小さい直



(a) (Dia., ASPL)=(2, 1.667) (b) (Dia., ASPL)=(3, 1.889)

図 1: Example of general graphs $(n, d) = (10, 3)$ [11]



(a) (Dia., ASPL)=(5, 3.144) (b) (Dia., ASPL)=(5, 3.162)

図 2: Example of grid graphs $(w \times h, d, r) = (6 \times 6, 3, 2)$

径と ASPL を持つ図 1a の方が図 1b よりも良いグラフである。次に、格子グラフ $(w \times h, d, r) = (6 \times 6, 3, 2)$ の例を図 2 に示す。グラフ中の赤線は距離が 2、黒線は距離が 1 を示している。頂点間のエッジがどれだけ長くても、そのホップ数は 1 と数えるため、一般グラフと同様の方法で直径と ASPL を計算することができる。図 2 においては、2 つのグラフで直径は同じであるが、より小さい ASPL を持つ図 2a の方が図 2b よりも良いグラフである。

一般グラフと格子グラフにおける直径と ASPL の理論的な下界は、それぞれ文献 [13] と文献 [14] で示されている。ただし、どちらのグラフも、それぞれの下界を満たすグラフの存在は保証されない点に注意が必要である。

2.2 Graph Golf

Order/Degree 問題を題材にした国際コンペティション Graph Golf が国立情報学研究所の主催で開催されている [15]。Graph Golf は年ごとに異なる頂点数・次数・最大長の組合せの問題が出題される。Graph Golf が開始された 2015 年は一般グラフのみの出題だったが、2017 年から格子グラフも出題されている。Graph Golf では、実在のスパコンに即した問題などが出題される。

Graph Golf では、各カテゴリにおいて「Widest Improvement Award」と「Deepest Improvement Award」という2つの賞が用意されており、最終ランキングを元にそれらは参加者に授与される。「Widest Improvement Award」は最良のトポロジを最も多く発見した投稿者に授与されるのに対し、「Deepest Improvement Award」は理論的な下界に最も近いトポロジを発見した投稿者に授与される。前者はアルゴリズムの汎用性の高さを評価するのに対し、後者は問題に特化したアルゴリズムの性能の高さを評価すると言える。我々は2018年の一般グラフ部門および2019年度の格子グラフ部門において、「Widest Improvement Award」と「Deepest Improvement Award」の両方の賞を受賞した。本稿で説明するアルゴリズムは、各部門で用いたアルゴリズムである。

2.3 関連研究

2.3.1 一般グラフ

Order/Degree 問題の一般グラフと類似の問題に Diameter/Degree 問題がある [16]。Diameter/Degree 問題とは、与えられた直径と次数を満たすグラフの集合の中から、最大の頂点数を持つグラフを発見する問題である。Diameter/Degree 問題は Order/Degree 問題よりも古くから研究が行われており、これまで発見されたグラフがまとめられている [17]。また、Diameter/Degree 問題で発見されたグラフをネットワークトポロジに応用する研究も行われている [18]。Diameter/Degree 問題で発見されるグラフの特徴として、その頂点数は飛び飛びの値をとることが挙げられる。並列計算機システムの計算ノード数は予算や電力などの制約で決定するため、Diameter/Degree 問題で発見されている既存のグラフをユーザが新規に作成するネットワークにそのまま適用できる可能性は低い。ただし、既存のグラフの頂点数が新規に作成するネットワークの頂点数に近い場合、既存のグラフの頂点を追加もしくは削除することにより、直径と ASPL が比較的小さいグラフを生成できることが報告されている [19,20]。

Order/Degree 問題で用いられている手法は、次の2つに分類できる。(1) 頂点数が小さい複数のグラフ同士を組合せる。(2) ランダムグラフを元に最適化アルゴリズムを用いる。(1)は直径と ASPL の計算回数が(2)と比べて少ないという利点を持つ反面、前述した Diameter/Degree 問題と同様に、特定の頂点数を持つグラフしか作れないという問題点がある。文献 [21] では、2種類のグラフを掛け合わせて新しいグラフを構成する Multiple star product という手法を提案している。文献 [22] では、複数の Petersen グラフを組合せることで、直径3のグラフを構成する手法を提案している。(2)は(1)とは逆であり、任意の頂点数のグラフを作成できる利点を持つ反面、直径と ASPL の計算を繰り返し行う必要があるため、その計算量が大きいと

いう問題点がある。文献 [23] は最適化アルゴリズムとして SA を用いており、また直径3のグラフに限定しているが、グラフの直径と ASPL を少ない計算量で近似する手法を開発している。本稿で述べる手法は、文献 [23] と同様に SA をベースとしているが、任意の直径のグラフを扱うことができる。また、SA の解探索能力に悪影響を与えないようにするため、グラフの直径と ASPL には正確な値を用いている。ただし、詳細は2.3.3節で説明するが、グラフに対称性を与えることにより、その直径と ASPL を求める計算時間の大幅な短縮を実現している [24]。

2.3.2 格子グラフ

格子グラフに関する研究は一般グラフに関する研究と比較して少ない。文献 [14] では、SA を用いた格子グラフの構成法を提案している。本稿で提案する格子グラフに関する手法は、文献 [14] を拡張したものである。4章において、我々の手法と文献 [14] との比較を行う。文献 [25] では、頂点同士のあるパターンに従って接続する格子グラフ作成法を提案している。ただし、 $(w \times h, d, r) = (N \times N, 5, 4)$ という特殊な正方格子グラフのみの対応である。

2.3.3 無向グラフにおける全点对間最短経路

グラフの直径と ASPL を正確に求めるためには、全頂点間の最短経路である全点对間最短経路 (APSP : All Pairs Shortest Path) を計算する必要がある。なお、Order/Degree 問題で扱うグラフは、一般グラフと格子グラフのどちらであっても重みなしグラフである。APSP アルゴリズムの中で最も有名かつ簡易に実装できるのはワーシャルフロイド法 [26,27] であるが、その計算量は $O(n^3)$ と大きい。他の APSP アルゴリズムに Seidel 法があり、その計算量は $O(n^{2.376} \log n)$ である [28]。ワーシャルフロイド法と Seidel 法は密なグラフの場合には有効であるが、本稿で対象とする Order/Degree 問題は工業製品などへの利用を想定しているため、比較的疎なグラフを扱うという特徴がある。以上のことから、Order/Degree 問題においては、ワーシャルフロイド法と Seidel 法は向いていないと言える。

我々の過去の研究 [24] では、隣接行列を用いた手法である ADJ-APSP の評価を行っている。ADJ-APSP の計算量は $O(n^2 dK/E)$ である。 K はグラフの直径であり、Order/Degree 問題で扱うグラフはスモールワールド効果により直径は比較的短いという特徴がある。 E は隣接行列の1要素のビット数であり、我々の実装では `uint64_t` を用いているため、 $E = 64$ である。さらに、3章で説明するグラフの対称性を利用することにより、その計算量を対称性の数に比例して減らすことができる。対称性の数を g とした場合、ADJ-APSP の計算量は $O(n^2 dK/Eg)$ になる。また、複数の GPU を用いた高速化も行っている。特に言及していないが、3章で説明する最適化アルゴリズムでは、この ADJ-APSP を内部的に用いている。

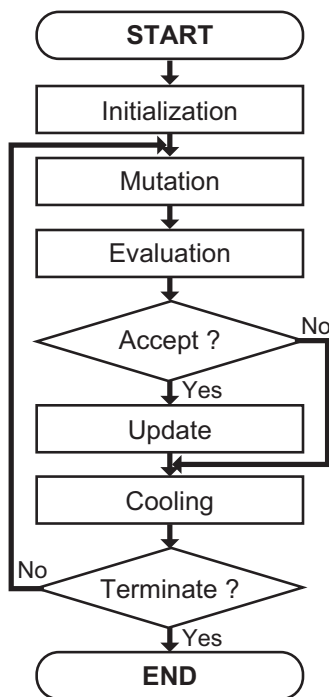
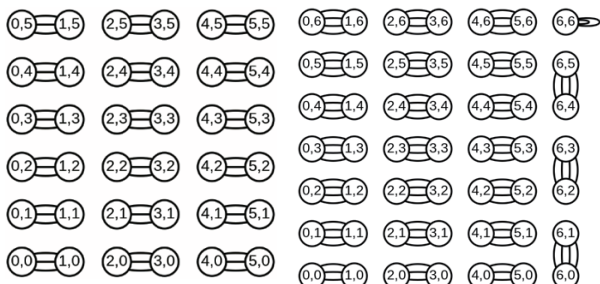


図 3: Flow of simulated annealing



(a) $(w \times h, d)=(6 \times 6, 3)$ (b) $(w \times h, d)=(7 \times 7, 4)$

図 4: Example of initial grid graphs

3. 提案手法

3.1 節では, SA をどのように Order/Degree 問題に適用するかについて説明する. 3.2 節では, 提案手法のキープポイントである対称性について説明する. 3.1 節と 3.2 節で述べる内容は, 格子グラフを除けば, 我々の過去の研究 [11] とほぼ同じである. 3.3 節では, 一般グラフにおける対称性の拡張について述べる.

3.1 SA への適用

図 3 に示す SA のフローチャートを用いて説明する. 各節のタイトルは, フローチャートの各項目と連携している.

3.1.1 Initialization

頂点同士をランダムに接続したランダムグラフを初期解として準備する. 初期解の作成方法はいくつかあるが, 本稿では Graph Golf 公式サイトで公開されているサンプル

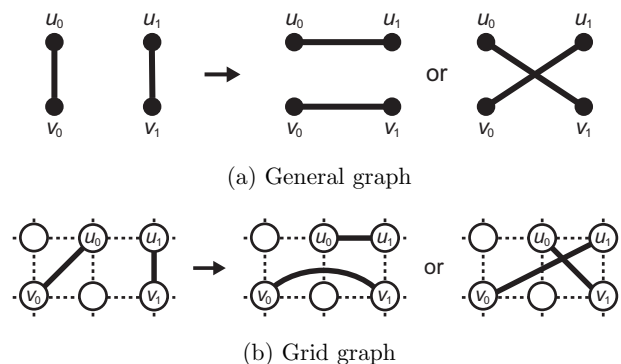


図 5: 2-opt method

コードを元にした方法を説明する.

一般グラフにおける初期解の生成には, Python パッケージである networkx [29] の random_regular_graph メソッドを利用する. このメソッドは, 指定した頂点数と次数を持つランダムグラフを出力する.

格子グラフにおける初期解の生成には, 文献 [14] で用いられている次の方法を用いる. (1) 図 4 に示すような, 隣接する頂点同士をつないだグラフを作成する. 図 4a は w と h が偶数の場合を示しており, 図 4b は w と h が奇数の場合を示している. もちろん, w と h で偶奇が異なっても良い. この時点ではエッジの最大長は 1 であるため, 最大長の制約 r は必ず満たす. (2) 3.1.2 節で述べる 2-opt 法を, 十分な回数 (1) のグラフに対して行う.

3.1.2 Mutation

現在のグラフから新しいグラフの生成を行うため, 図 5 に示す 2-opt 法を用いる. 2-opt 法では, 現在のグラフから 2 本のエッジをランダムに選択し, それらを入れ替えることで新しいグラフを生成する. なお, 入れ替え方は 2 通りあるため, そのどちらかをランダムに選択する. ただし, 格子グラフの場合は, 最大長の制約条件を満たす場合のみ 2-opt を行う. もし入れ替えた結果, 最大長の制約条件を満たさない場合は, エッジの選択からやり直す.

3.1.3 Evaluation

SA は新しいグラフ (x_{new}) と現在のグラフ (x_{cur}) が持つ評価値の差分 ΔE を必要とする. 評価値には APSL を用いる. ΔE を導く式を次に示す. $E(x)$ はあるグラフ (x) の APSL を表している.

$$\Delta E = E(x_{new}) - E(x_{cur}) \quad (1)$$

ここで, 2.1 節で説明した距離行列と APSL の定義により, ΔE の最小値は $2/(n^2 - n)$ である. すなわち, 頂点数が多いほど ΔE の最小値は小さくなる. 最小値が問題毎に異なると温度パラメータの設定が煩雑になるため, ΔE に重みとして $(n^2 - n)$ を掛けた $\Delta E'$ を用いることにする.

$$\Delta E' = \Delta E \times (n^2 - n) \quad (2)$$

```

1 function bool accept(...)
2   if new_diam < cur_diam then
3     return true
4   else if new_diam > cur_diam then
5     return false
6   end if
7
8   if new_ASPL <= current_ASPL then
9     return true
10  else
11    diff = (new_ASPL-cur_ASPL)*n*(n-1)
12    if exp(-diff/T) > uniform_rand() then
13      return true
14    else
15      return false
16    end if
17  end if
18 end function

```

図 6: Acceptance condition

3.1.4 Acceptance Condition

3.1.2 節で作成した新しいグラフを受理するか否かの判定を行う。その擬似コードを図 6 に示す。Order/Degree 問題では、ASPL よりも直径が優先される。そこで、まず 2~6 行目では、新しいグラフの直径 (*new_diam*) と現在のグラフの直径 (*cur_diam*) を比較し、新しいグラフの直径の方が小さい場合は、ASPL の値に関わらず受理する設計にした。逆に、新しいグラフの直径の方が大きい場合は、ASPL の値に関わらず受理しない。8 行目以降 (新しいグラフと現在のグラフの直径が同じ場合) は、 $\Delta E'$ および温度 T と次の Metropolis 基準を用いて、新しいグラフを受理するか否かの判定を行う。

$$Probability = \begin{cases} 1 & \text{if } \Delta E' < 0 \\ \exp(-\frac{\Delta E'}{T}) & \text{otherwise} \end{cases} \quad (3)$$

Metropolis 基準では、新しいグラフの ASPL (*new_ASPL*) が現在のグラフの ASPL (*cur_ASPL*) よりも小さい場合は、新しいグラフを必ず受理する。その逆の場合は、新しいグラフを確率的に受理する。温度が高いほど、その受理確率は高くなる。

3.1.5 Update

3.1.4 節で受理の場合は、現在のグラフを新しいグラフに置換する。

3.1.6 Cooling

次の指数型アニーリングを用いて、温度 T を下げる処理を行う。 α はクーリング率であり、最高温度 T_{max} 、最低温度 T_{min} 、イテレーション回数 N とすると、 $\alpha = (T_{min}/T_{max})^{1/N}$ である。

$$T \leftarrow \alpha \times T \quad (4)$$

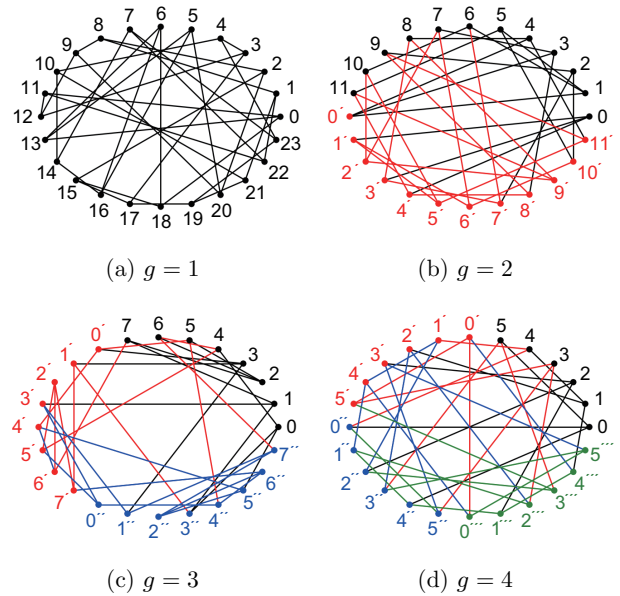


図 7: General graphs with symmetry (n, d) = (24, 3) [11]

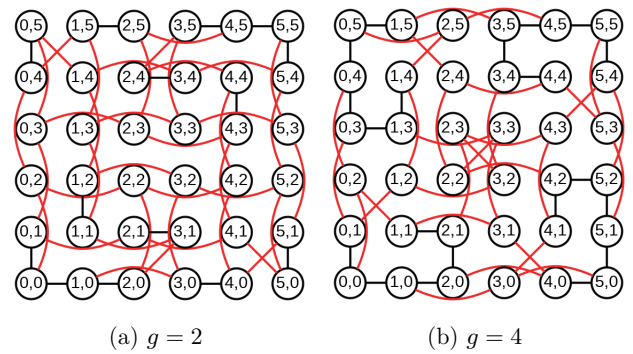


図 8: Grid graphs with symmetry ($w \times h, d, r$) = (6x6, 3, 2)

3.1.7 Terminal Condition

イテレーションが、初期パラメータであるイテレーション回数 N に達したら、プログラムを終了する。なお、図 3 では省略しているが、3.1.3 節で説明した“Evaluation”の処理において、新しいグラフの直径と ASPL が共に下界である場合もプログラムを終了する。

3.2 グラフの対称性

3.2.1 概要

Order/Degree 問題が難しい理由として、その探索空間が非常に広いこと、直径と ASPL の計算に膨大な時間を要することが挙げられる。そこで、グラフに対称性を与えることにより、探索空間を削減し、直径と ASPL の計算時間を短縮する工夫を行う。

対称性を与えた一般グラフの例を図 7 に示す。図中の変数 g は対称性の数であり、各グラフを平面としてみた場合、 $360/g$ 度回転させると頂点とエッジの接続関係が同じグラフになる。なお、 $g = 1$ の場合は、対称性を持たない通常

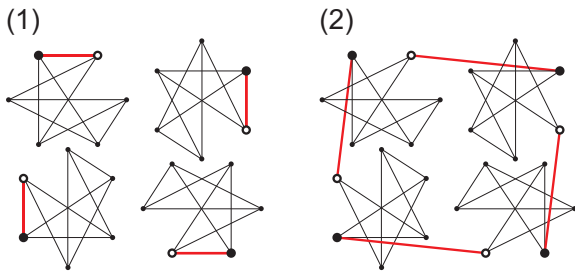


図 9: Generating initial graph with symmetry [11]

のグラフである。同様に、対称性を与えた格子グラフの例を図 8 に示す。この定義により、対称性の数 g は頂点数 n の約数である必要がある。さらに、格子グラフでは、 g の取りうる値は 1, 2, 4 のみであり、 $g = 4$ の場合は、 $w = h$ の正方格子グラフである必要がある。

次の 3.2.2~3.2.4 節では、3.1 節で説明した SA に対して対称性をグラフに与える機能を追加する。具体的には、初期解として対称性を持つグラフを準備し、その対称性を保ったまま、最適化を行っていく。これらの機能は、一般グラフと格子グラフで同様の手法を用いて実現できるため、一般グラフのみを例にして説明する。3.2.2~3.2.4 節以外の処理は、3.1 節で述べた処理と同じである。

3.2.2 Initialization

一般グラフ $(n, d, g) = (24, 3, 4)$ を例として、初期解を作成する。その概要を図 9 と次に示す。(1) 作成したいグラフの頂点数 n を対称性の数 g で割った値 ($6 = n/g$) を頂点数として持つグラフを 3.1.1 節に従い作成する。次数 d は対象となるグラフと同じである。このグラフをベースグラフと呼ぶ。ベースグラフを g 個複製した後、各ベースグラフからエッジを対称的に 1 つずつ選択する。便宜上、図 9 では、4 つのベースグラフは 90 度ずつ回転させており、また選択したエッジの始点 \circ と終点 \bullet を設定している。(2) エッジの始点 \circ と別のベースグラフのエッジの終点 \bullet とを接続する。

3.2.3 Mutation

一般グラフ $(n, d, g) = (12, 3, 2)$ を例として、グラフの対称性を保ったまま、新しい解を生成する手法について述べる。その概要を図 10 と次に示す。(1) すべてのエッジの中からランダムに 2 つのエッジを選択する。(2) 選択した 2 つのエッジと対称的な関係にあるエッジを選択する。(3) (1) で選択したエッジ間で 2-opt 法を行う。(4) (3) と同様の 2-opt 法を (2) で選択したエッジ間でも行う。

次に、上記の 2g-opt 法が行えない例を図 11 と次に示す。図 10 の (1) で選択した 1 つ (もしくは 2 つとも) のエッジの両端が対称的な関係である場合 (図 11a)、もしくは図 10 の (1) で選択した 2 つのエッジが対称的な関係である場合 (図 11b)、2-opt 法を行うことができない。このような場合、選択済みの 2 つのエッジから 1 つを選択し、そのエッジに対して図 9 に示したように別のベースグラフ

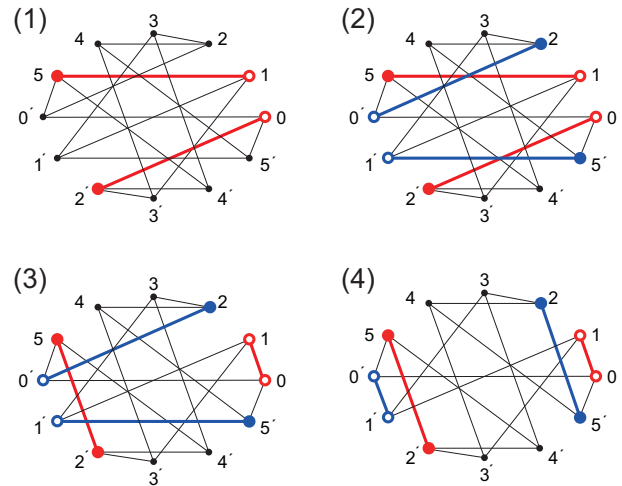
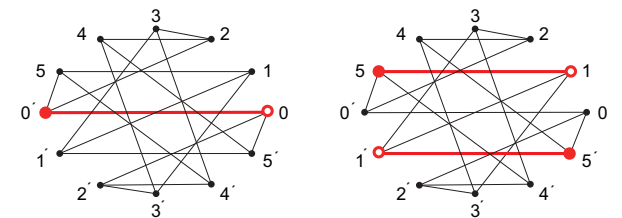


図 10: Overview of 2-opt method with symmetry [11]



(a) Both ends of selected edge are symmetric (b) Two selected edges are symmetric

図 11: Cases where 2-opt method with symmetry cannot be performed [11]

に接続する操作を行うことで、新しいグラフを作成する。

3.2.4 Evaluation

グラフの対称性を利用することにより、その直径と ASPL を求めるための計算量を削減することが可能である。一般に、グラフの直径と ASPL を求めるためには、すべての頂点から他の頂点までのホップ数を計算する必要がある。しかしながら、対称性を持つグラフでは、対称的な関係のある頂点同士において、その頂点から他の頂点までのホップ数はすべて同じであるという性質がある。例えば、図 7d では、頂点 $0, 0', 0'', 0'''$ から、他の頂点までのホップ数の集合はすべて同じである。そのため、図 7d では、頂点 $0 \sim 5$ から他の頂点までのホップ数を計算するだけでグラフの直径と ASPL を求めることができる。すなわち、対称性を持つグラフの直径と ASPL を求めるには、 n/g 個の頂点から他の頂点までのホップ数を計算するだけで良い。

ここで予備実験を行った結果、図 10 に示す 2-opt 法は図 5 に示す通常の 2-opt 法の g 倍のエッジ交換を行うため、評価値の差 $\Delta E'$ の変化量が g に比例して大きくなることがわかった。式 (3) より、 $\Delta E'$ と温度 T は受理率に関係している。ある 1 つの問題において、 g の値を変える毎に、温度 T も調整することは煩雑な作業である。そこで、式 (5) のように $\Delta E'$ に重みとして $1/g$ を掛けた $\Delta E''$ を評

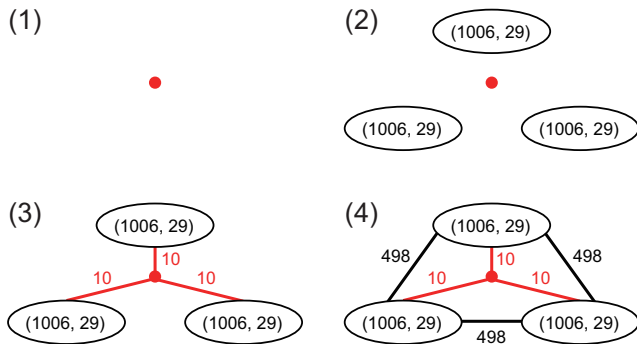


図 12: Creating graph with prime vertices and symmetry

価値の差分として用いることにより、同じ問題であれば g の値に依らずに同じ温度を設定できるようにした。

$$\Delta E'' = \Delta E' / g = \Delta E \times (n^2 - n) / g \quad (5)$$

3.3 一般グラフにおける中心点を用いた対称性の拡張

3.2.1 節で述べたように、対称性の数 g は頂点数 n の約数である必要がある。そのため、頂点数が素数の場合は、対称性を利用することができない。そこで、本節では、その対称性を中心点を用いて拡張することにより、素数の頂点数を持つ一般グラフに対応する方法について述べる。なお、格子グラフの頂点数は素数にならないため、本節では取り扱わない。

中心点を伴う対称性の作成方法を次に説明する。(1) c 個の点をグラフの中心に配置する。この点を中心点と呼ぶ。(2) ベースグラフを中心点の周りに配置する。各ベースグラフの頂点数は $(n-c)/g$ であり、次数は $d-1$ である。次数が $d-1$ である理由は、後の処理でエッジを追加するためである。(3) 各ベースグラフの適当な頂点と中心点を結ぶ新しいエッジを追加する。中心点の次数を d にするため、中心点に結ばれる各ベースグラフ内の頂点数は cd/g になる。(4) 各ベースグラフの中心点と接続しない頂点同士を結ぶ新しいエッジを追加する。各ベースグラフの中心点と接続しない頂点の数は $(n-c-cd)/g$ である。

例として、2018 年の Graph Golf で出題された素数の頂点数を持つグラフ $(n, d) = (3019, 30)$ を用いて説明する。その概要を図 12 と次に示す。(1) 1 つの中心点を配置する。(2) $(n, d) = (1006, 29)$ のベースグラフを 3 つ配置する。(3) 次数が 30 であるため、各ベースグラフから 10 個の頂点を選択し、それらと中心点とを接続する。(4) 各ベースグラフ間で、(1) で選択しなかった残りの 996 個の頂点同士を接続する。この場合、996 個の半分である 498 個の頂点同士が、隣接するベースグラフ間で接続することになる。以上の操作により、 $(n, d, g, c) = (3019, 30, 3, 1)$ のグラフが完成する。他には、 $(n, d, g, c) = (3019, 30, 5, 9)$ や $(3019, 30, 15, 19)$ などが作成可能である。

この中心点を伴う対称性の制約条件は次の通りである。

(1) ベースグラフの頂点数 $(n-c)/g$ は自然数である。(2) 握手の補題により、ベースグラフの頂点数 $(n-c)/g$ もしくは次数 $(d-1)$ のどちらかは偶数である。(3) 中心点の次数は d であるため、対称性の数 g が取りうる値は次数 d の約数になる。(4) 各ベースグラフの中心点と接続しない頂点は隣のベースグラフと接続する必要があるため、 $(n-c-cd)/g$ は偶数である。

4. 提案手法の解探索性能の評価

本章では対称性の数 g と SA の解探索性能の関係について調べる。SA のパラメータであるイテレーション回数 N は 10^6 とした。最高温度 T_{max} と最低温度 T_{min} は、我々の過去の研究 [11] に従い、下記の通り設定した。

最高温度 初期解から 2-opt 法を 100 回行った中の最大の評価値の差分 $\Delta E''$ を 50% の確率で受理する値。

最低温度 最小の評価値の差分 $\Delta E''$ を 0.01% で確率で受理する値。

4.1 一般グラフの評価

2018 年と 2019 年の Graph Golf で出題された問題である $(n, d) = (512, 4), (2300, 10), (9344, 6), (64K, 6), (200k, 32), (1m, 32)$ の 6 つを用いる。K は 2^{10} , k は 10^3 , m は 10^6 である。各問題における 10 試行中の最良値の結果を図 13 に示す。各値の丸括弧内の数字は直径である。また、比較のためにランダムグラフの各値も示している。

この結果より、提案手法の直径と ASPL はランダムグラフのそれらよりも良い結果を示すことがわかる。また、基本的に対称性の数が多いほど良い結果を得られることもわかる。ただし、対称性の数が多すぎると、SA の性能が劣化することもわかる。その理由は、対称性の数が大きくなるほど、そのグラフの規則性が強くなるため、スモールワールド性が弱くなるからと考えられる。

提案手法の最良の g における APSL の改善率 i ($= (1.0 - \text{提案手法の APSL} / \text{ランダムグラフの APSL}) \times 100$) は下記の通りである。 $(n, d, i) = (512, 4, 5.14), (2300, 10, 1.65), (9344, 6, 1.27), (64K, 6, 0.56), (200k, 32, 0.13), (1m, 32, 0.19)$ 。このことから、小さい問題ほど改善率が高くなることがわかる。その理由は、すべての問題において評価計算回数 N は同じであるため、小さい問題の方が最適化がより強く行えたからであると考えられる。

4.2 格子グラフの評価

2019 年の Graph Golf で出題された問題である $(w \times h, d, r) = (20 \times 20, 4, 2), (20 \times 20, 6, 3), (20 \times 20, 8, 4), (100 \times 100, 4, 2), (100 \times 100, 6, 3), (100 \times 100, 8, 4)$ の 6 つを用いる。各問題における 10 試行中の最良値の結果およびランダムグラフの値を図 14 に示す。なお、 $g = 1$ の結果は、文献 [14] のアルゴリズムを用いた場合と同じである。

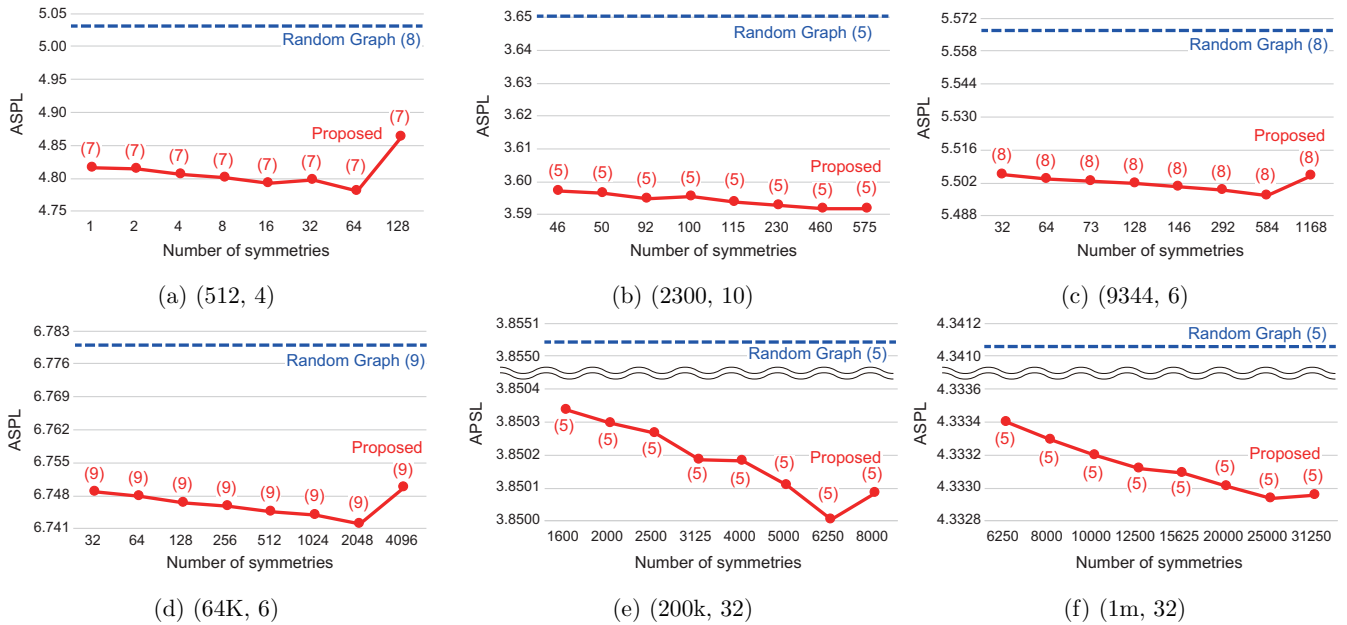


図 13: Result of general graphs (n, d)

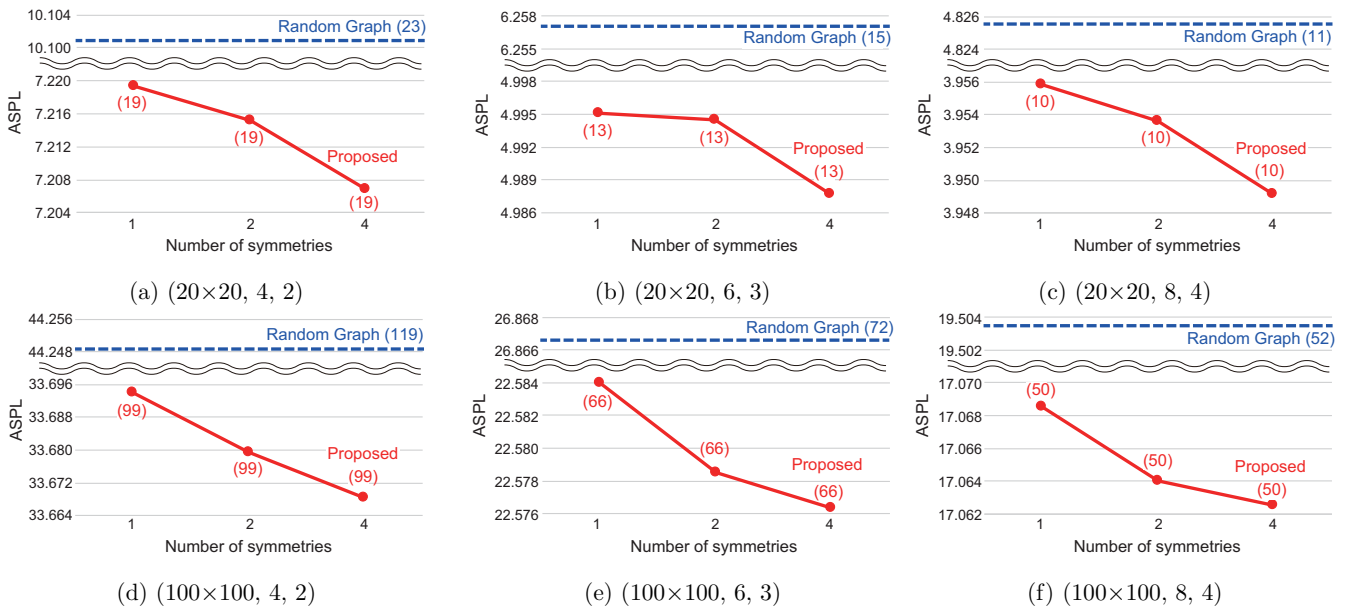


図 14: Result of grid graphs ($w \times h, d, r$)

この結果より、4.1節で述べた一般グラフの場合と同様に、提案手法の直径と ASPL はランダムグラフよりも良い値であり、さらに対称性の数が多いほどよくなることからわかる。提案手法の最良の g における ASPL の改善率 i は下記の通りである。($w \times h, d, r, i$) = (20×20, 4, 2, 28.65), (20×20, 6, 3, 20.29), (20×20, 8, 4, 18.17), (100×100, 4, 2, 23.19), (100×100, 6, 3, 15.97), (100×100, 8, 4, 12.52)。同じ頂点数 $w \times h$ の問題においては、次数 d と最大長 r が小さい問題ほど改善率が高くなることからわかる。また、格子グラフにおける ASPL の改善率は、一般グラフの改善率より大きいことがわかる。言い換えると、格子グラフの方が一般グラフよりも、最適化の余地が大き

いことがわかる。この理由としては、格子グラフでは距離の制約条件があるため、距離の制約条件が厳しいほど、スモールワールド性が弱くなるからと考えられる。

4.3 素数の頂点数を持つ一般グラフの評価

3.3節で述べた頂点数が素数であるグラフ (n, d) = (3019, 30) を用いて提案手法の評価を行う。評価に用いる対称性の数と中心点の数は、(g, c) = (1, 0), (3, 1), (5, 9), (15, 19) である。(g, c) = (1, 0) は、中心点を用いない $g = 1$ の場合を指す。10 試行中の最良値の結果およびランダムグラフの値を図 15 に示す。

この結果より、提案手法の ASPL はランダムグラフより

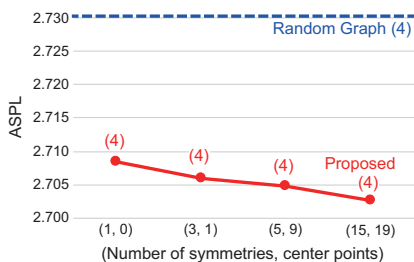


図 15: Result of general graph with prime vertices $(n, d)=(3019, 30)$

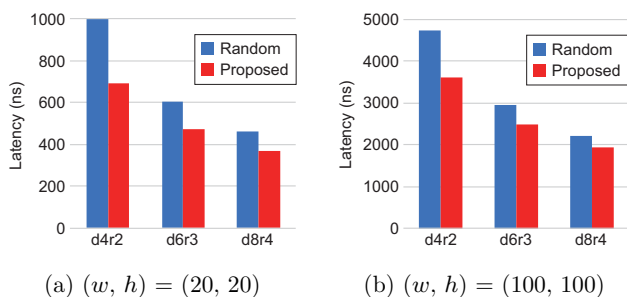


図 16: Average latency in grid graphs

も良い値であり、対称性の数が多いほどより良くなること
がわかる。提案手法の最良の g における ASPL の改善率 i
は 1.01 である。

5. 生成トポロジの評価

5.1 格子グラフにおけるレイテンシ

図 14 で作成した格子グラフにおけるネットワークのエ
ンドポイント間のレイテンシを計算する。簡略化のため、
格子の 1 マスは $1m \times 1m$ とし、各節点上に 1 つのエ
ンドポイント (e.g. スイッチングハブ) があるとす。エ
ンドポイントの遅延を 100ns, ケーブルの遅延を 5ns/m とする。

全エンドポイント間の最短経路をダイクストラ法を用い
て計算した。その最短経路を用いて、すべてのエ
ンドポイントから他のエンドポイントまで 0 バイトの通信を行
ったと仮定した場合の平均レイテンシを図 16 に示す。この結
果より、ランダムトポロジよりも最適化を行ったトポロジの
方が小さいレイテンシであることがわかる。 $g = 4$ のレイ
テンシの大きさはランダムトポロジの 69.10%~88.11%
であった。また、最大長が小さい問題ほど、改善率が高いこ
とがわかる。この理由は、4.2 節で述べた通り、最大長が
小さい問題におけるランダムグラフの直径と ASPL は比較
的大きくなるため、最適化したグラフのそれらとの差が大
きくなるからである。

また、 $g = 4$ の場合の定性的なメリットとしては、対
称性がある方が $g = 1$ の場合 [14] やランダムトポロジより
もケーブルの複雑性が小さくなると考えられるため、施工や
保守の際のヒューマンエラーを減らすことができると考え
られる。

表 1: Results in $(n, d)=(512, 6)$ and $(512, 8)$

	$(512, 6)$		$(512, 8)$	
	Random	Proposed	Random	Proposed
Diameter	6	5	5	4
ASPL	3.76	3.60	3.27	3.14

5.2 並列アプリケーションの性能

並列シミュレーションフレームワーク SimGrid-3.24 [12]
を用いて、提案アルゴリズムで生成した各グラフにお
ける並列アプリケーションの性能測定を行う。SimGrid の設
定では、各計算ノードの性能、ネットワークのバンド幅、
スイッチングハブのレイテンシはそれぞれは 200GFlops,
100Gbps, 100ns に設定した。また、MPI 集合通信の設定に
は、MVAPICH2 implementation を選択した。並列ア
プリケーションには、SimGrid に同梱されている NPB3.3-MPI
(NPB) [30] の BT, CG, DT, FT, IS, LU, MG, SP の
CLASS B を用いた。各アプリケーションの説明と主な通
信パターンは下記の通りである [4]。

BT 3 重対角行列の連立 1 次方程式ソルバー。隣接通信。

CG 共役勾配法。1 対 1 通信。

DT データトラフィック。1 対 1 通信。

FT 3 次元の高速フーリエ変換。All-to-All 通信。

IS 整数ソート。All-to-All 通信。

LU LU 分解を用いた連立 1 次方程式ソルバー。隣接通信。

MG マルチグリッド計算。短距離と長距離の 1 対 1 通信。

SP 5 重対角行列の連立 1 次方程式ソルバー。隣接通信。

一般グラフの評価には、図 13a に示す $(n, d) = (512, 4)$
と、追加して $(512, 6)$ と $(512, 8)$ を用いる。 $(512, 6)$ と
 $(512, 8)$ については、4.1 章と同様の方法を用いて最適
化を行った。 $(512, 6)$ と $(512, 8)$ の最良の対称性の数は共
に 128 であった。表 1 に、最適化を行った各グラフの直径と
ASPL を示す。それぞれの改善率 i は、 $(n, d, i) = (512, 4, 5.14)$,
 $(512, 6, 4.00)$, $(512, 8, 3.79)$ であるので、同じ頂
点数であれば、次数 d が小さいほど ASPL の改善率は高
くなることわかる。この理由は、次数が小さいほど疎な
グラフになるため、最適化の余地がより大きくなるからと
考えられる。格子グラフの評価には、図 14a~図 14c に示
す $(w \times h, d, r) = (20 \times 20, 4, 2)$, $(20 \times 20, 6, 3)$, $(20 \times 20, 8, 4)$
を用いる。

ランダムグラフの性能値を 1.0 として最良の g を用い
て生成したグラフの性能値を正規化した結果を図 17 に示
す。値が大きいほど良い性能である。この結果から、す
べてのベンチマークにおいて、ランダムグラフよりも最適
化されたグラフの方が優れていることがわかる。なお、
一般グラフ (図 17a~図 17c) における最適化された
グラフを用いた場合の平均値は 1.07, 1.04, 1.02 であり、
格子グラフ (図 17d~図 17f) における最適化された
グラフを用いた場合の平均値は 2.18, 2.08, 1.65 であ
った。このことから、

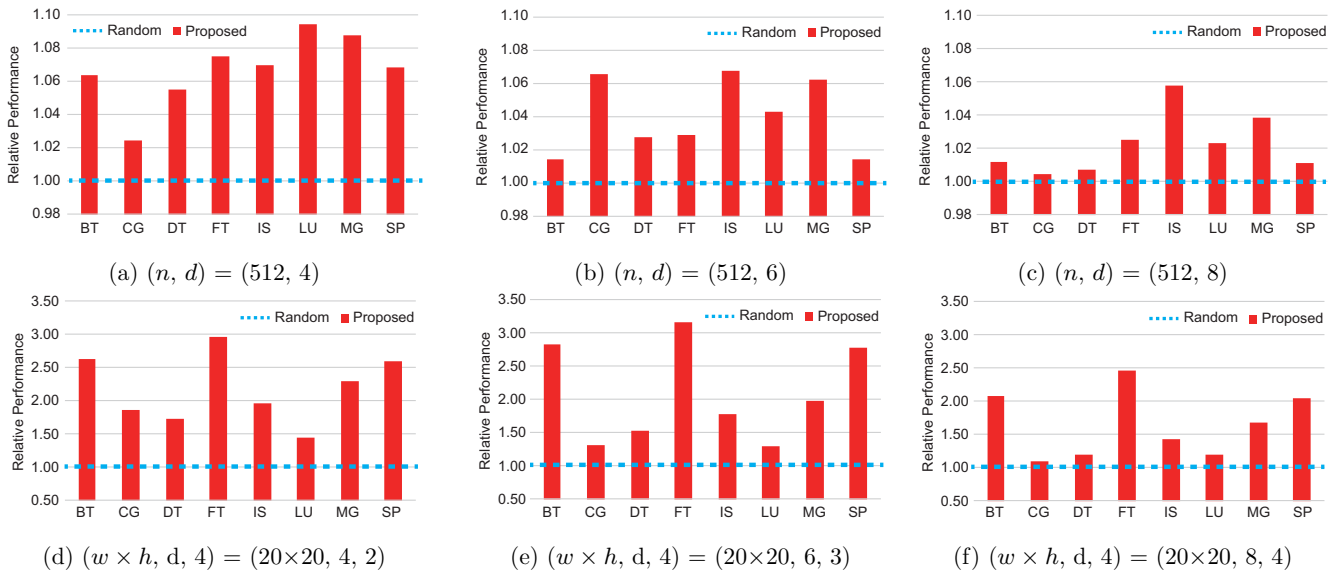


図 17: Performance of NAS Parallel Benchmarks on SimGrid

次数と最大長が小さいほど、ランダムグラフとの差が大きくなることわかる。

6. まとめと今後の課題

本稿では、低遅延な相互結合網を設計するため、グラフ理論の Order/Degree 問題に対する最適化アルゴリズムの提案を行った。そのアルゴリズムの特徴として、解に対称性を持たせることで、より低遅延な相互結合網を生成できることが挙げられる。さらに、頂点を自由に配置できる一般グラフと、頂点を 2 次元格子の節点上に配置する格子グラフの両方に適用できる点が挙げられる。本稿の評価において、頂点数が最大 100 万の大規模グラフを用いた評価を行った。その結果、高い対称性を持つグラフの方がより小さい直径と ASPL を持つことを明らかにした。さらに、設計したグラフに対し、レイテンシの測定と並列ベンチマークを用いたシミュレーションを行った結果、ランダムグラフよりも最適化されたグラフの方が高い性能を発揮できることを示した。また、ランダムグラフや対称性を持たないグラフと比較して、対称性を持つケーブルの方が複雑性が小さくなるため、施工性や保守性においても有利であると考えられる。

今後の課題として、下記を挙げることができる。(1) 本稿で提案した最適化アルゴリズムは SA をベースにしているが、Genetic Algorithm (GA) [31,32] などの他の最適化アルゴリズムを用いることができる。SA は 1 つの解に対して摂動を与える近傍探索であるため、解は少しずつしか良くならない。それに対し、GA は複数の解を組合せて新しい解を生成するため、SA よりも少ない評価計算回数で良い解を発見できる可能性がある。(2) 格子グラフにおいて、本稿で提案した手法における最大の対称性の数は 4 であり、同じ頂点数を持つ一般グラフと比較してその対称性

の数は少ない。より対称性の数を増やすためには、対称性の定義を拡張する必要がある。例えば、格子グラフにおける本稿で提案した対称性は点対称をベースとしている。対称性の定義を線対称に変更することで、2 のべき乗の対称性を定義することができる。(3) 実際の並列計算機システムのネットワークは、長い光ケーブルと短いメタルケーブルの両方が用いられることが多い。そして、光ケーブルの方がメタルケーブルよりも高価である。そこで、より現実に即した Order/Degree 問題として、指定されたコスト内で光ケーブルとメタルケーブルの両方を用いることができるルールにすると面白いかもしれない。(4) 規則的な通信パターンを持つアプリケーションは、その通信の規則性と同じトポロジを持つネットワークが最適である。例えば、3 次元ステンシル計算を行うアプリケーションにおける最適なトポロジは k -ary 3-cube である。しかしながら、実アプリケーションにおいては、隣接ノードが不規則である場合や複数の通信パターンが必要である場合があるため、最適なネットワークトポロジが容易にはわからない場合が多い。また、予算やハードウェア上の仕様の都合で、理想的なネットワークトポロジを設計できない場合もある。そこで、アプリケーションの通信パターンを実行時に記録しておき、ネットワークに対する制約条件を本提案アルゴリズムを組込むことで、そのアプリケーションに特化したネットワークトポロジを生成できると考えている。

Acknowledgements

This research used the Cygnus system provided by Interdisciplinary Computational Science Program in the Center for Computational Sciences, University of Tsukuba. This work was supported by JSPS KAKENHI Grant Number 18K11331 and RIKEN Incentive Research Projects. We are grateful to the Graph Golf committee for giving us the interesting issues.

参考文献

- [1] Hiroki Matsutani et al. Low-latency wireless 3D NoCs via randomized shortcut chips. In *2014 Design, Automation Test in Europe Conference Exhibition (DATE)*, pp. 1–6, March 2014.
- [2] Ankit Singla et al. Jellyfish: Networking Data Centers Randomly. *CoRR*, Vol. abs/1110.1687, , 2011.
- [3] Ji-Yong Shin et al. Small-world Datacenters. In *Proceedings of the 2Nd ACM Symposium on Cloud Computing*, SOCC '11, pp. 2:1–2:13, New York, NY, USA, 2011. ACM.
- [4] F. Chaix, I. Fujiwara, and M. Koibuchi. Suitability of the random topology for hpc applications. In *2016 24th Euromicro International Conference on Parallel, Distributed, and Network-Based Processing (PDP)*, pp. 301–304, Feb 2016.
- [5] Michihiro Koibuchi et al. A case for random shortcut topologies for HPC interconnects. In *2012 39th Annual International Symposium on Computer Architecture (ISCA)*, pp. 177–188, June 2012.
- [6] M. Koibuchi, I. Fujiwara, H. Matsutani, and H. Casanova. Layout-conscious random topologies for hpc off-chip interconnects. In *2013 IEEE 19th International Symposium on High Performance Computer Architecture (HPCA)*, pp. 484–495, Feb 2013.
- [7] I. Fujiwara, M. Koibuchi, H. Matsutani, and H. Casanova. Swap-and-randomize: A method for building low-latency hpc interconnects. *IEEE Transactions on Parallel and Distributed Systems*, Vol. 26, No. 7, pp. 2051–2060, July 2015.
- [8] Michihiro Koibuchi et al. Optical network technologies for HPC: computer-architects point of view. *IEICE Electronics Express*, Vol. 13, No. 6, pp. 1–14, 2016.
- [9] Nicholas Metropolis et al. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, Vol. 21, No. 6, pp. 1087–1092, 1953.
- [10] Scott Kirkpatrick et al. Optimization by simulated annealing. *Science*, Vol. 220, No. 4598, pp. 671–680, 1983.
- [11] Masahiro Nakao et al. A method for order/degree problem based on graph symmetry and simulated annealing with mpi/openmp parallelization. In *Proceedings of the International Conference on High Performance Computing in Asia-Pacific Region, HPC Asia 2019*, pp. 128–137. ACM, 2019.
- [12] Henri Casanova, Arnaud Giersch, Arnaud Legrand, Martin Quinson, and Frédéric Suter. Versatile, scalable, and accurate simulation of distributed applications and platforms. *Journal of Parallel and Distributed Computing*, Vol. 74, No. 10, pp. 2899–2917, June 2014.
- [13] V. G. Cerf, D. D. Cowan, R. C. Mullin, and R. G. Stanton. A lower bound on the average shortest path length in regular graphs. *Networks*, Vol. 4, No. 4, pp. 335–342, 1974.
- [14] K. Nakano and D. Takafuji and S. Fujita and H. Matsutani and I. Fujiwara and M. Koibuchi. Randomly optimized grid graph for low-latency interconnection networks. In *45th International Conference on Parallel Processing (ICPP)*, pp. 340–349, Aug 2016.
- [15] Michihiro Koibuchi et al. Graph Golf: The Order/degree Problem Competition, 2018. <http://research.nii.ac.jp/graphgolf>.
- [16] Mirka Miller and Jozef vSirvn. Moore graphs and beyond: A survey of the degree/diameter problem. *Electronic Journal of Combinatorics, Dynamic survey*, Vol. 14, pp. 1–61, 12 2005.
- [17] The Degree/Diameter Problem - Combinatorics Wiki. http://combinatoricswiki.org/wiki/The_Degree/Diameter_Problem.
- [18] M. Besta and T. Hoefler. Slim fly: A cost effective low-diameter network topology. In *SC '14: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 348–359, Nov 2014.
- [19] M. Koibuchi, I. Fujiwara, F. Chaix, and H. Casanova. Towards ideal hop counts in interconnection networks with arbitrary size. In *2016 Fourth International Symposium on Computing and Networking (CANDAR)*, pp. 188–194, Nov 2016.
- [20] Teruaki KITASUKA, Takayuki MATSUZAKI, and Masahiro IIDA. Order adjustment approach using cayley graphs for the order/degree problem. *IEICE Transactions on Information and Systems*, Vol. E101.D, No. 12, pp. 2908–2915, 2018.
- [21] Ryosuke Mizuno and Yawara Ishida. Constructing large-scale low-latency network from small optimal networks. In *2016 Tenth IEEE/ACM International Symposium on Networks-on-Chip (NOCS)*, pp. 1–5, Aug 2016.
- [22] T. Kitasuka and M. Iida. A heuristic method of generating diameter 3 graphs for order/degree problem (invited paper). In *2016 Tenth IEEE/ACM International Symposium on Networks-on-Chip (NOCS)*, pp. 1–6, Aug 2016.
- [23] Nobutaka Shimizu and Ryuhei Mori. Average Shortest Path Length of Graphs of Diameter 3. *CoRR*, Vol. abs/1606.05119, , 2016.
- [24] Masahiro Nakao, Hitoshi Murai, and Mitsuhisa Sato. Parallelization of all-pairs-shortest-path algorithms in unweighted graph. In *Proceedings of the International Conference on High Performance Computing in Asia-Pacific Region, HPCAsia2020*, pp. 63–72, New York, NY, USA, 2020. ACM.
- [25] Fujita Satoshi, Nakano Koji, Koibuchi Michihiro, and Fujiwara Ikki. Deterministic construction of regular geometric graphs with short average distance and limited edge length. In *Algorithms and Architectures for Parallel Processing*, pp. 295–309, Cham, 2016. Springer International Publishing.
- [26] Stephen Warshall. A Theorem on Boolean Matrices. *J. ACM*, Vol. 9, No. 1, pp. 11–12, January 1962.
- [27] Robert W. Floyd. Algorithm 97: Shortest Path. *Commun. ACM*, Vol. 5, No. 6, p. 345, June 1962.
- [28] Raimund Seidel. On the All-Pairs-Shortest-Path Problem in Unweighted Undirected Graphs. *Journal of Computer and System Sciences*, Vol. 51, No. 3, pp. 400–403, 1995.
- [29] NetworkX developers. NetworkX, 2018. <https://networkx.github.io>.
- [30] Bailey D.H. et al. THE NAS PARALLEL BENCHMARKS. Technical Report NAS-94-007, Nasa Ames Research Center, 1994.
- [31] John H. Holland. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*. MIT Press, Cambridge, MA, USA, 1992.
- [32] David E. Goldberg. Genetic Algorithms in Search Optimization and Machine Learning. *AI Magazine*, Vol. 12, pp. 102–103, 1989.