

# 小直径グラフ探索コンペ “Graph Golf” 5年間の成果

藤原 一毅<sup>†</sup> 中野 浩嗣<sup>††</sup> 鯉渕 道紘<sup>†,††</sup>

<sup>†</sup> 国立情報学研究所 〒101-8430 東京都千代田区一ツ橋 2-1-2  
<sup>††</sup> 総合研究大学院大学 〒101-8430 東京都千代田区一ツ橋 2-1-2  
<sup>††</sup> 広島大学 〒739-8527 東広島市鏡山 1-4-1  
E-mail: <sup>†</sup>{ikki,koibuchi}@nii.ac.jp, <sup>††</sup>nakano@cs.hiroshima-u.ac.jp

あらまし Graph Golf は、筆者らが開催しているプログラミングコンテストの一種で、グラフの Order/Degree 問題の解の最適性を競うものである。Order/Degree 問題は、与えられた頂点数と次数をもつグラフの直径・平均距離を最小化する問題であり、この問題の効率的な解法を得ることは、大規模並列計算機の低遅延な相互結合網の設計につながると期待される。筆者らは Graph Golf を 2015 年から毎年 1 回開催しており、これまでに累計 1000 件を超える投稿を受け付けて、いくつかの興味深い知見を得た。本報告では、Graph Golf を通じてこれまでに得られた成果を総括するとともに、並列計算機の相互結合網への応用に向けた展望を述べる。

キーワード ネットワークトポロジ, 相互結合網, グラフ理論, Order/Degree 問題

## Five-year achievement of the Graph Golf competition

Ikki FUJIWARA<sup>†</sup>, Koji NAKANO<sup>††</sup>, and Michihiro KOIBUCHI<sup>†,††</sup>

<sup>†</sup> National Institute of Informatics 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo, 101-8430 Japan  
<sup>††</sup> SOKENDAI (The Graduate University for Advanced Studies) 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo, 101-8430 Japan  
<sup>††</sup> Hiroshima University 1-4-1 Kagamiyama, Higashihiroshima-shi, Hiroshima, 739-8527 Japan  
E-mail: <sup>†</sup>{ikki,koibuchi}@nii.ac.jp, <sup>††</sup>nakano@cs.hiroshima-u.ac.jp

### 1. はじめに

プロセッサのメニーコア化、計算機システムの大規模並列化（チップ内：100 コア、メモリネットワーク：1,000 チップ、チップ間：100 万ノード規模）が進むにつれて、計算機システムのメモリ、ストレージ、プロセッサコアなどの構成要素間を接続するネットワークである相互結合網の通信遅延が計算システム全体に与える性能の割合が大きくなってきている。例えば、次世代のスーパーコンピュータ（以後、スパコンと呼ぶ）では、アプリケーションが、システム全体で数百ナノ秒～1 マイクロ秒という極めて小さい MPI 通信遅延を必要とすることが予測されている [1]。また、そのメニーコアプロセッサチップ内のパケットネットワークに関しては、プロセッサコア間通信遅延およびラストレベルキャッシュとの通信遅延が十サイクル程度であることが求められており、チップ内のコア数の増大がその実現をより困難なものとしている。最近では、大容量メモリを搭載するために、ボード上に 3 次元積層したメモリキューブを多数配置し、パケットネットワークで相互接続する技術が登場し

ているが、同様に、低通信遅延が必須である。

これらの設計では、コア間をどのように効率的に相互接続すべきか？というネットワーク構成（以下、ネットワークトポロジと呼ぶ）が極めて重要である。相互結合網の通信遅延を削減する有効な 1 つの方法は、直径・平均距離の小さいネットワークトポロジを採用することである。実際に、直径・平均距離の小さなネットワークトポロジを採用することにより、多くの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 [2], [3]。

与えられた次数・直径を満たす中で、頂点数がなるべく大きいグラフを見つける問題は Degree/Diameter 問題と呼ばれ、これまでに多くのグラフ理論家に取り組んできた成果が蓄積されている [4]。しかし、計算機システムのノード数は予算や電力などの外部要因で決まるため、Degree/Diameter 問題の解をそのままポロジとして採用できる可能性は低い。

相互結合網を設計する上では、与えられた頂点数・次数を満たす中で直径がなるべく小さい無向グラフを見つけることが求められる。これは Order/Degree 問題と呼ばれる。Order/Degree

問題は、有向グラフについては任意の頂点数、次数に対して、ムーア限界あるいは高々1ホップ大きい直径のグラフを生成する効率的な手法が存在する [5].

しかし、大規模並列計算機の相互結合網は双方向リンクであり、無向グラフとしてモデル化される。我々の知る限り、無向グラフの Order/Degree 問題の効率的な解法はあまり研究されておらず、有向グラフのような優れたグラフ構成法があるのか? という点もふくめて既知ではない。

そこで我々は、グラフ理論研究者による研究の進展を待つのではなく、オープンサイエンスの手法によって Order/Degree 問題の追究を加速することを目指して、小直径グラフ探索コンペ “Graph Golf” を開催している。このコンテストは、直径・平均距離の小さいグラフとその構成法を広く一般から募集し、優れたグラフの作者を国際会議 CANDAR (The Seventh International Symposium on Computing and Networking) [6] で表彰するものである。

コンペについては積極的にアナウンス [7]~[10] を行い、幅広く参加者を募り、一定の注目を浴びた [11]~[13]。特に、コンペの表彰者の方々の全面的なご協力により、国際会議第10回 IEEE/ACM International Symposium on Networks-on-Chip (NOCS 2016) の GraphGolf 特別セッションや第15回情報科学技術フォーラム (FIT2016) の企画イベントにおいてこれまでに得られた知見を積極的に発信し、第22回夏の電脳甲子園 (SuperCon2016) で関連問題が出題されるなど、優れたグラフを発見する取り組みは新たな広がりを見せた。コンペの年毎の簡単な報告は [14]~[21] をご欄いただきたい。

本報告は、はじめに GraphGolf の概要を述べる (2. 章)。次に、GraphGolf コンペの一般部門 (3. 章)、格子グラフ部門 (4. 章) の詳細を紹介する。最後に、応用上の展望 (5. 章) とまとめ (6. 章) を述べる。

## 2. Graph Golf の概要

Graph Golf の開催に至った背景と Order/Degree 問題の詳細については、文献 [22] で報告した通りである。ここでは問題の定義を簡潔に述べ、コンペ開催の経緯を振り返る。

### 2.1 Order/Degree 問題

本報告では単純無向グラフを扱う。グラフ上の2頂点を結ぶ最短経路上の辺数=ホップ数をグラフ上の距離とする。あるグラフ  $G$  について、 $G$  上のすべての2頂点の組合せに対する距離の最大値を直径 (diameter)、平均値を平均距離 (ASPL: average shortest path length) と呼ぶ。Order/Degree 問題 (ODP) は、与えられた頂点数 (order)  $n$  と次数 (degree)  $d$  をもつグラフ<sup>(注1)</sup>のなかで、直径  $k$  が最小であるグラフを探す最適化問題である。直径が等しいグラフが複数あるときは、それらのうち平均距離  $l$  が最小であるグラフを探すものとする。

2016年までは一般の単純無向グラフのみを扱っていたが、2017年からは新たに格子グラフ部門を設けた。格子グラフ

部門の出題では、2次元ユークリッド空間上で長方形をなす  $n = w \times h$  個の格子点を頂点とし、次数を  $d$ 、辺の長さを  $r$  以下とする制約が与えられる (辺の長さはマンハッタン距離で測る)。例えば  $w = 2, h = 3, d = 2, r = 2$  の場合、 $2 \times 3$  の長方形形状に並んだ6個の頂点を長さ2以下の辺で結んだ次数2のグラフが受理される。なお、辺の長さはグラフ上の距離に影響せず、長さの異なる辺もすべて1ホップと数えて直径・平均距離を計算する。この制約は、配線長に上限があり斜め配線が許容されないチップ内ネットワークやデータセンターを想定したものである。

### 2.2 コンペの進行

Graph Golf では、主催者が出題したいいくつかの頂点数  $n$  と次数  $d$  の組合せに対し、ODP の解となるべきグラフを参加者が探して投稿する。参加資格は特に決めておらず、オンライン投稿システムが利用できれば誰でも参加できる。投稿受付期間は4~5か月と長く、例年5月下旬ごろ受付開始、10月上旬ごろ締切としている。前半はクローズド投稿期間、後半はオープン投稿期間であり、オープン投稿期間中は投稿されたグラフが毎週月曜日に公開される。参加者は他の参加者の解と比較して自分の順位を確認したり、他の参加者の解をベースとしてさらなる改良に取り組んだりできる。2019年のコンペは5月13日受付開始、7月22日からオープン投稿期間、10月14日締切の予定である。

投稿されたグラフは次の2つの基準で評価される。ひとつは、それぞれの  $(n, d)$  ごとに、グラフの直径  $k$  と平均距離  $l$  から導かれるスコア  $score = 1000000k + l$  が最も小さいグラフを最善解とする基準であり、最多数の最善解を見つけた参加者に Widest Improvement Award が与えられる。もうひとつは、すべての  $(n, d)$  にわたって、グラフの平均距離  $l$  と下界  $L_{n,d}$  とのギャップ  $gap = l - L_{n,d}$  が最も小さいグラフを最善解とする基準であり、最善解を見つけた参加者に Deepest Improvement Award が与えられる。なお、スコアやギャップが同点となる場合、最も早い週に公開されたグラフのみが最善解と見なされる。受賞者は例年、国際会議 CANDAR 併設の Graph Golf ワークショップに招待され、授賞式と記念講演が行われる。過去の講演資料はホームページ<sup>(注2)</sup>で公開されている。

### 2.3 運営システム

Graph Golf は web 上で投稿受付と順位発表を行うオンラインコンペである。この種のオンラインコンペを運営するための汎用システムとして CodaLab<sup>(注3)</sup>などが存在するが、我々は独自の運営システムを開発した。図1にシステム構成を示す。本システムは、投稿受付用のアップローダをパブリッククラウドに、グラフ評価およびHTML生成部を研究室所有の計算クラスターに、静的ウェブサイト国立情報学研究所のwebサーバに、それぞれ配備した。このようなハイブリッド構成とした理由は、グラフの直径・平均距離の計算量が大きいこと

(注2) : <http://research.nii.ac.jp/graphgolf/>。以下「ホームページ」と書いた場合はこのサイトを指す。

(注3) : <https://codalab.lri.fr/>。GraphGolf2019は<https://competitions.codalab.org/competitions/22858> においても公開中。

(注1) : 正則グラフに限定されない。頂点ごとの次数の最大値をグラフの次数とする。

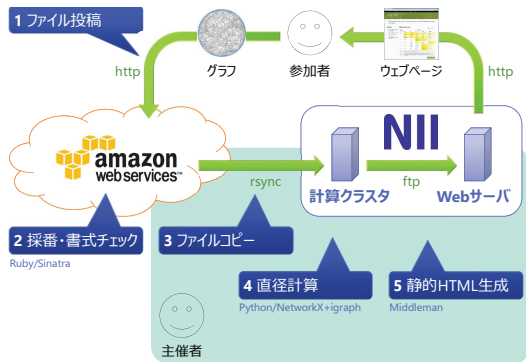


図1 システム構成

と、webサーバがポリシーにより動的サイトをホストできないことによる。アップローダはRuby+Sinatra+Nginxで、グラフ評価システムはPython+NetworkXで、静的ウェブサイトはRuby+Middlemanで、それぞれ開発した。アップローダはAmazon EC2のt2.mediumインスタンスで稼働している。

コンペ開催中、ユーザーがグラフファイル（エッジリスト形式のテキストファイル）をアップローダに投稿する。アップローダは投稿されたグラフファイルをパースし、フォーマットが適切か否か、頂点数・次数が出題に合致しているか否かを判定し、否であればユーザーに確認を求め、適切であればグラフとメタデータをファイルとして保存する。なお、出題に合致していないグラフも投稿でき、ホームページに掲載されるが、表彰対象とならない。

週に一度の更新時、運営者が計算クラスタ上で更新コマンドを手動実行する。更新コマンドはグラフとメタデータのファイルをアップローダからローカルクラスタにコピーし、未計算のグラフファイルに対して直径・平均距離を計算し、メタデータファイルを更新する。その後、メタデータファイル内の情報に基づいてHTMLファイル群を生成し、webサーバにアップロードする。直径・平均距離の計算には当初、NetworkXの組み込み関数を利用していたが、頂点数1万を超えるグラフを扱うには計算時間が長すぎたため、現在は2015年入賞者の森氏が開発したプログラム<sup>(注4)</sup>を利用している。このプログラムを利用すると、Xeon E5-2667のシングルスレッドで、100万頂点・32次のランダムグラフの直径・平均距離を118分程度で計算できる。

### 3. 一般グラフ部門

#### 3.1 背景

スーパーコンピュータ（スパコン）に代表される並列計算機は大規模化・高並列化が進み、数万の計算ノードからなるシステムも珍しくない。スパコンを構成するノード群は通常、ひとつのマシナールーム内に配置され、光ケーブルまたは電気ケーブルで接続される。ケーブル中を伝搬する信号の速さが光速の2/3程度（一般的な光ファイバの場合）である一方、ノード（ネットワークスイッチ）がパケットを中継するには数百マ

表1 実在のスパコンに即した出題

	頂点数	次数	想定マシン	Top500 (2017/11)
2018年	3019	30	Cori	8位
2018年	4855	30	Trinity	7位
2017年	4896	24	Tianhe-2A	2位
2017年	9344	10	Titan	5位
2017年	88128	12	K Computer	10位
2017年	98304	10	Sequoia	6位

イクロ秒の時間がかかることから、ノード間の通信遅延は途中で経由するノード数（スイッチ数）によって支配されていると言える。つまり、通信遅延の小さい並列計算機とは、平均的に経由ノード数が少ないネットワークトポロジをもつシステムである。このようなシステムをモデル化するには、ケーブルの実延長とは無関係に、グラフ上の距離＝ホップ数の平均値に注目することが妥当である。したがって我々は、一般グラフ部門におけるOrder/Degree問題の評価値を、前章で述べたように $score = 1000000k + l$ と定義した。

#### 3.2 問題設定

一般グラフ部門の出題一覧を付録表A.1に示す。これらのパラメータを選んだ意図として、以下の類型がある。

- (1) 入門用の小規模問題 ( $n \leq 36$ )
- (2) 実在のチップマルチプロセッサに即した問題 ( $n = 72$ )
- (3) 実在のスパコンに即した問題 (表1)
- (4) スパコンのロードマップなどで報告された将来のスパコンに即した問題 [1], [23]
- (5) 高い計算能力を要する大規模問題 ( $n \geq 100000$ )
- (6) Degree/Diameter問題 [4] の最善解の更新を狙うもの ( $n = 2300$  など)
- (7) グラフ理論の未解決問題に挑むもの ( $n = 3250$ )
- (8) キリの良い数 ( $n = 256, 1024$  など)

このうち(5) Degree/Diameter問題 (DDP) とは、与えられた次数と直径をもつグラフの中で頂点数が最大のグラフを探す問題であり、既知の最善解がカタログ化されている [4]。また(6) グラフ理論の未解決問題とは、頂点数3250、次数57、直径2のMoore graphが存在しうるものの、いまだ発見されていないことを指す。もしGraph Golfの参加者がこれらのグラフを発見すれば、グラフ理論分野で大いに注目されるだろう。

#### 3.3 直径・平均距離の下界

正則グラフにおいて、ある頂点から距離*i*で到達できる頂点数の上界をMoore boundといい、本報告では $M(i)$ と書く。次数 $d \geq 2$ の正則グラフ*G*に対するMoore boundは

$$M(i) = \begin{cases} 1 & \text{if } i = 0 \\ 1 + d \sum_{j=1}^i (d-1)^{j-1} & \text{if } i \geq 1 \end{cases} \quad (1)$$

で与えられる。*G*の平均距離の下界 $L_{n,d}$ は、*G*のひとつの頂点から他の $n-1$ 個の頂点までの距離の下界の平均であるから、

$$L_{n,d} = \frac{\sum_{i \geq 1} ((M(i) - M(i-1)) \cdot i)}{n-1} \quad (2)$$

となる。また、直径の下界 $K_{n,d}$ は

(注4) : [https://github.com/ryuhei-mori/graph\\_ASPL](https://github.com/ryuhei-mori/graph_ASPL)

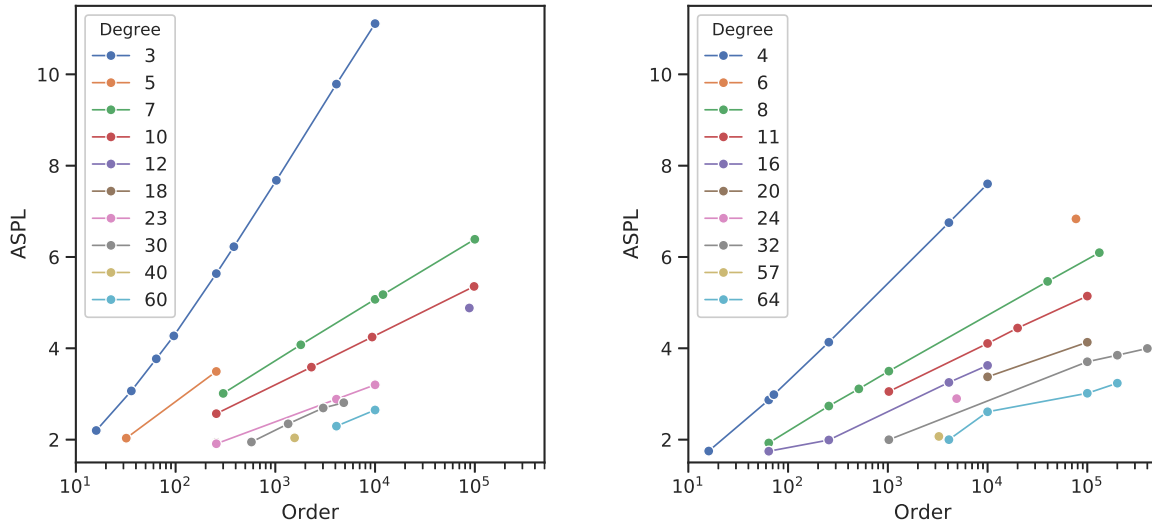


図 2 一般グラフ部門の最善解の平均距離を次数ごとに示す。プロットが重なり合わないよう 2 つの図に分けた。

$$K_{n,d} = \begin{cases} \lceil \frac{(n-1)}{2} \rceil & \text{if } d = 2 \\ \lceil \log_{d-1} \left( \frac{(n-1)(d-2)}{d} + 1 \right) \rceil & \text{if } d > 2 \end{cases} \quad (3)$$

となる。

### 3.4 解法

#### a) ベースライン

ランダムグラフの直径・平均距離は規則的なグラフに比べて非常に小さくなることが経験的に知られている。そこで、各出題の条件を満たすランダムグラフを主催者が生成し、初期解として投稿受付開始と同時に公開した。生成には NetworkX の `random_regular_graph` 関数を用いた。このプログラムはホームページで公開されている。

#### b) 2-opt とメタヒューリスティクス

ODP の解を探す最も素朴な方法は、グラフ上の 2 辺を任意に選んで互いの頂点を入れ替え (= を × にする)、直径・平均距離が小さくなれば確定、さもなければもとに戻す、という操作を繰り返す方法である。この操作を 2-opt と呼ぶ。しかしながら、ODP は探索空間が大きすぎて、このようなローカルサーチで最適解に到達することは絶望的と言える。そこで、多くの参加者は simulated annealing (SA) などのメタヒューリスティクスを用いて探索を行った。

#### c) 平均距離の計算を高速化する

頂点数  $n$ 、次数  $d$  のグラフの平均距離の計算量は、各頂点から幅優先探索を行うと  $O(n^2d)$  となる。この計算量が大きいため、特に頂点数の大きいグラフにおいて、良い解を現実的な時間内に見つけれないことが問題となる。そこで、平均距離の計算を高速化するために参加者はさまざまな工夫を凝らした。

2015 年に入賞した清水らは、直径 3 のグラフに限り、2-opt による平均距離の変化量を  $O(1)$  で概算するアルゴリズムを開発した [24]。2016 年に参加した井上は、Path Count Index と

称する索引表を持つことで、2-opt による平均距離の変化量を高速に推定するとともに、その推定値を次に 2-opt を行うべき辺の選択に利用した。2018 年に入賞した中尾は、グラフに対称性を持たせることで計算量を削減するとともに、MPI と OpenMP によるハイブリッド並列化を行い、頂点数 400000、次数 32 のグラフの平均距離の計算時間をおよそ  $1/2000000$  に短縮した [25]。

#### d) 既知の小直径グラフを組み合わせる

2015 年に入賞した水野らは、2 種類のグラフを掛け合わせて ODP の最適解となるグラフを構成する Multiple star product という手法を開発した [26]。同じく入賞した北須賀らは、直径 3 のグラフにターゲットを絞り、直径 2 の Petersen グラフを複数結合して要件を満たすグラフを構成した [27]。2016 年に再び入賞した水野らは、任意の整数  $N$  に対し、直径 3、頂点数  $2^{2N}$ 、次数  $2^N$  のグラフを構成する手法を開発した。2017 年に入賞した川又は、Voltage graph という構成法を用いた。同じく入賞した松崎は、複数の Cayley graph を Brown's construction という手法で組み合わせた後、要件を満たすよう辺を増減してグラフを構成した。

### 3.5 成果

コンペを通じて得られた最善解の平均距離を図 2 に示す。また、最善解を含むすべてのデータを付録表 A.1 に示す。

$(n, d) = (16, 3), (32, 5), (64, 16), (72, 4), (256, 10), (256, 23), (576, 30)$  において、直径・平均距離とも理論下限に等しいグラフが発見された。このうち  $(n, d) = (72, 4)$  は Intel Xeon Knights Landing のチップ内ネットワークに即した出題であり、実在の製品に対して改善の可能性を示すことができた。

実在のスパコンに即した出題においても、例えば 2017 年 11 月の Top500 ランキングで 8 位になった米国 NERSC の Cori スーパーコンピュータに即した問題  $(n, d) = (3019, 30)$  で直径

が理論下限に等しいグラフを発見するなど、ネットワークのさらなる低遅延化を図る余地があることを示した。

さらに、出題外の任意投稿による成果として、グラフ理論分野の有名問題である Degree/Diameter 問題の既知の最善解（最も頂点数が大きい解）を上回る頂点数をもつグラフが4個発見された。それらは  $(n, d) = (2394, 10), (20468, 11), (80050, 6), (137745, 8)$  である。

## 4. 格子グラフ部門

### 4.1 背景

計算機ネットワークの一部の設計では、以下の通り、厳しい配線長制約が課されてきた。

#### a) スパコン、データセンターネットワーク

InfiniBand に代表される計算ノード間ネットワークは、長距離配線用にアクティブ光ケーブル、短距離用にパッシブ電気ケーブルが用いられている。2010 年ころまで、前者は高価、後者は 10-40Gbps リンクの場合、10m 長のサポート可能であった。そのため、スパコンのネットワークに電気ケーブルを多用するためにフロアレイアウト時の配線長を抑える設計が有効であった。ただし、現在は、(1) アクティブ光ケーブルが安価で品質が劇的に向上した、(2) 100-200Gbps の広帯域化によりパッシブ電気ケーブルは、長さが 3m にとどまり、事実上ラック内あるいは隣接ラック間配線の利用に限定される、という2点から、多数の（配線長に制約の小さい）アクティブ光ケーブルを用いる設計が増えてきている。

なお、配線長は、2つの計算ノード間の直線距離ではなく、マシルーム（2次元平面）上の X 方向の距離と Y 方向の距離の和となる。つまり、マンハッタン長となる。また、マシルームは正方形とは限らず、長方形となる場合も多々存在する。

#### b) メモリネットワーク

メモリは容量は制約されるがプロセッサの近くに統合積層する HBM (High Bandwidth Memory) と、大容量化が可能である DDR や Memory Cube が挙げられる。後者は、ボードを介して通信するため、プロセッサからの距離が HBM と比べて長い。そのため、例えば、HMC (Hybrid Memory Cube) において数十 Gbps のリンクを実装する場合、リンクの消費電力が数ワットとなり、メモリ本体の消費電力に匹敵する。そのため、総リンク数が小さくなるように、HMC のネットワークの次数は 4 と小さい。そのため、多数の HMC をボード上に設置する場合、配線長を抑えるボード上のレイアウトが望ましい。なお、本配線長も、マンハッタン長となる。また、HMC は、ボード上の正方形のエリアに設置されるとは限らず、長方形となる場合も想定される。

#### c) チップ内ネットワーク (Network on Chip)

最近のプロセッサチップは、コア間、コアオンチップキャッシュ間通信にパケット転送を用いるチップ内ネットワークを採用している。チップ内ネットワークでは、最長配線長を抑えることで、(1) リピーターを排除した消費電力の低減、(2) リンク遅延を抑制、かつ、チップの動作周波数の向上、に寄与することが期待できる [28]。なお、本配線長も、マンハッタン長であ

表 2 3つのネットワークの特徴

	スパコン NW	メモリ NW	チップ内 NW
配線長	~数十 m	~数十 cm	~数 mm
ノード数	~10 万	~1000	~100
短配線の利点	電気ケーブル	省電力	省電力/高周波数

表 3 格子グラフ部門の出題の分類

	頂点数	次数	辺長	主な想定 NW
2016 年	16~256	3~15	2~15	チップ NW
2017 年	10,000	3~28	6~33	スパコン NW
2018 年	64~1024	4	3~24	メモリ NW

る。また、最近のプロセッサチップは、正方形であることは稀で、通常、長方形である。

TSV (Through-Silicon Via) を用いてチップを 3次元垂直統合積層することが可能となりつつある。そのため、チップ内ネットワークを 3次元拡張する研究が盛んに行われている。しかし、3次元方向の通信はグラフゴルフが対象とするパケットネットワークではなく、Dynamic Time Division Multiple Access (D-TDMA) パスを用いることが多い。そのため、3次元チップ内ネットワークは本報告の対象外である。

以上、3つのネットワークの特徴を表 2 にまとめる。

### 4.2 問題設定

グラフゴルフでは、前節で述べた短距離配線を実現する、優れたネットワークポロジを発見するために、2017 年のコンペティションから格子グラフ部門を設けている。格子グラフ部門では、2次元平面上に頂点と辺があるグリッドグラフにおいて、頂点が格子状に並んでおり、かつ、平面上の辺長に制約を満たすグラフを競う。

出題は、直径と平均最短経路長の下界（次節参照）が正則グラフの下界（ムーア限界）が支配的な場合、辺長による下界が支配的な場合、両方の下界に等しく依存している場合の3パターンから各々選択している。それぞれの出題は表 2 のネットワークの少なくとも1つに合致するように考えられている（表 3）。格子グラフ部門の出題一覧を付録表 A.2 に示す。

### 4.3 直径と平均距離の下界

頂点数  $\sqrt{n} \times \sqrt{n}$ 、次数  $d$ 、辺長  $r$  制約の格子グラフ  $G$  における直径と平均距離の下界は、次数  $d$  の正則グラフ（辺長無制限）の同下界と、辺長  $r$  制約の格子グラフ（次数無制限）の同下界を別々に求め、それらを統合することで求められる [29]。

格子グラフ  $G = \{V, E\}$  の頂点  $(x, y)$  から  $i$  ホップで到達できる頂点数  $U_{x,y}(i)$  は、頂点  $(x_1, y_1)$  から  $(x_2, y_2)$  までの辺長（マンハッタン距離）を  $T((x_1, y_1), (x_2, y_2))$  と書けば、

$$U_{x,y}(i) = \left| \{(x', y') \in V \mid T((x, y), (x', y')) \leq i \cdot r\} \right| \quad (4)$$

で与えられる。したがって、頂点数  $n$ 、辺長  $r$  制約の格子グラフ（次数無制限）の平均距離の下界  $L_{n,r}^U$  は

$$L_{n,r}^U = \frac{\sum_{(x,y) \in V} \sum_{i \geq 1} ((U_{x,y}(i) - U_{x,y}(i-1)) \cdot i)}{n(n-1)} \quad (5)$$

となる。

次数と辺長の両方に制限がある場合、つまり、次数  $d$  かつ辺長  $r$  制約の格子グラフの平均距離の下界は、両下界の最大値  $\max(L_{n,d}, L_{n,r}^U)$  となることは自明である。

ここで、さらに厳しい下界を求めるために  $M(i)$  と  $U_{x,y}(i)$  を統合し、 $MU_{x,y}(i) = \min(M(i), U_{x,y}(i))$  とする。次数  $d$ 、辺長  $r$  制約の格子グラフにおいて、頂点  $(x, y)$  から  $i$  ホップで到達できる頂点数は  $MU_{x,y}(i)$  を越えない。よって、下界  $L$  は次式となる。

$$L = \frac{\sum_{(x,y) \in V} \sum_{i \geq 1} ((MU_{x,y}(i) - MU_{x,y}(i-1)) \cdot i)}{n(n-1)}.$$

同様に、 $MU$  を用いて直径の下界を求めることができる。ここで、 $K$  を、 $MU_{0,0}(i-1) < n$  かつ  $MU_{0,0}(i-1) = n$  である値  $i$  とする。頂点  $(0, 0)$  からの距離が  $K$  個の辺を持つ頂点  $v$  が存在するため、 $K$  が直径の下界となる。

#### 4.4 解法

##### a) ベースライン

一般グラフ部門と同様に、主催者が生成したランダムグラフを初期解とした。ランダムグラフの生成法は次のとおり。

(1) 与えられたノード数と次数をもつ規則的な格子グラフを生成する。

(2) グラフ上の 2 辺をランダムに選んで互いの頂点を入れ替え、辺長制約が満たされるならば確定、さもなければもとに戻す、という操作を繰り返す。

このプログラムはホームページで公開されている。

##### b) 2-opt と高速化

格子グラフ部門は出題された頂点数が巨大でなかったこともあり、多くの参加者が一般グラフと同様の 2-opt にもとづくローカルサーチを採用したようである。

2017 年に入賞した川又は、2-opt による平均距離の変化量を  $O(ne)$  ( $e$  は辺数) で概算するアルゴリズムを開発して格子グラフ部門に適用した。

#### 4.5 成果

コンペを通じて得られた最善解を付録表 A.2 に示す。

### 5. 応用上の展望

ランダム性やスモールワールド性を用いて直径を小さくするネットワークポロジの研究がスパコン、データセンターネットワーク [30], [31]、メモリネットワーク [32]、チップ内ネットワーク [2], [33] に関して各々多数存在し、実装技術、消費電力、ルータのハードウェア量、性能などの面で議論されてきた。その結果、グラフゴルフを通して発見されたグラフをネットワークポロジとして採用する実装面での課題は大きくないと考えられる。一方、依然、Fat ツリー、トーラス、ドラゴンフライ (階層構造を持つ直接網) がスパコン、データセンター、チップ内ネットワークの実ネットワークで採用、運用されている。

グラフゴルフで発見されたグラフが不採用である理由として、(1) ホップ数の改善が与えるシステム性能の向上度合いが、新ネットワークポロジを採用する決断を後押しするに見合わないと現場で判断されている、(2) 古典並列計算アプリケーションが、ツリー構造、あるいはトーラスネットワーク構造に合致するように (十分なコストをこれまでに投じて) 最適化されてきた、という 2 つを推測している。

ただし、後者については、今後急速に変化することが予想される。グラフ上のサーチや機械学習などのビッグデータ処理、MapReduce 実装など最近 10 年以内に利用が拡大したアプリケーションでは、一元的なアクセス、いわゆる、ユニフォームトラヒックが生じやすい [3]。ユニフォームトラヒックの場合、平均最短経路長が小さいネットワークポロジを採用することで (ネットワークが混雑していない限り)、平均通信遅延を最小化できる。その点より、グラフゴルフを通して発見されたグラフをネットワークポロジに利用した並列計算機は、今後有望といえる。

### 6. おわりに

本報告では、複雑なネットワーク構成をスイッチ間の接続関係を表す簡単なグラフに抽象化し、より単純な構成のグラフの発見を競うコンペティション “Graph Golf” について報告した。代表的な成果を以下にまとめる。

- 2015 年から毎年 1 回の開催を継続し、これまでに累計 1000 件を超えるグラフを収集した。

- 一般グラフ部門では、 $(n, d) = (16, 3), (32, 5), (64, 16), (72, 4), (256, 10), (256, 23), (576, 30)$  において、直径と平均距離の面で理論的に最適なグラフが発見された。

- 最近の TOP500 ランキングの上位のスパコンに即した出題をした結果、直径 2~7 と極めて効率的なネットワークを構成できるグラフが発見された。つまり、実在のスパコンのネットワークの直径を縮めることに成功した。

- 投稿された 4 つのグラフがグラフ理論分野において著名な問題とされてきた Degree/Diameter 問題の最大グラフの記録を更新した。

- 与えられたグラフに対して、高速な平均最短経路長の高速計算アルゴリズムの開発を促進し、参加者が開発した高速計算プログラムを運営側が応募グラフの検算に利用するなど、グラフゴルフは参加者と共創に成功していると考えられる。

我々は、グラフゴルフの成果が、通信遅延を削減することを重視する最近のスパコンや計算機プロセッサチップのネットワーク構成の設計への直接的な利用につながることを期待している。本コンペは問題の条件設定を変えて今後も継続する予定で、グラフ (ネットワーク構成) のカタログを “Graph Bank”<sup>(注5)</sup> に蓄積していくことで学術界や産業界に貢献していきたい。

#### 文 献

- [1] K.S. Hemmert, J.S. Vetter, K. Bergman, C. Das, A. Emami, C. Janssen, D.K. Panda, C. Stunkel, K. Underwood, and S. Yalamanchili, “Report on IAA Interconnection Networks Workshop 2008,” Technical report, Future Technologies Group, Oak Ridge National Laboratory, 2008.
- [2] H. Matsutani, M. Koibuchi, I. Fujiwara, T. Kagami, Y. Take, T. Kuroda, P. Bogdan, R. Marculescu, and H. Amano,

(注5) : <http://graphbank.org>

- “Low-latency wireless 3D NoCs via randomized shortcut chips,” Design, Automation & Test in Europe Conference & Exhibition (DATE), 2014, pp.1–6, March 2014.
- [3] F. Chaix, I. Fujiwara, and M. Koibuchi, “Suitability of the random topology for HPC applications,” Proc. 24th Eurromicro Int. Conf. Parallel, Distributed, and Network-Based Processing, pp.301–304, 2016.
- [4] “The Degree/Diameter Problem - Combinatorics Wiki,” [http://combinatoricswiki.org/wiki/The\\_Degree/Diameter\\_Problem](http://combinatoricswiki.org/wiki/The_Degree/Diameter_Problem).
- [5] M. Imase and M. Itoh, “A Design for Directed Graphs with Minimum Diameter,” IEEE Trans. Computers, vol.32, no.8, pp.782–784, 1983.
- [6] “International Symposium on Computing and Networking, CANDAR,” 2014-2018. <https://is-candar.org/>
- [7] 国立情報学研究所ニュースリリース, “効率的なネットワーク構成を示すグラフ発見を競うコンペを今年も開催～専門家以外でも「理想のスパコンネットワーク」実現に貢献可能～,” <https://www.nii.ac.jp/news/release/2016/0609.html>, June 2016.
- [8] 国立情報学研究所ニュースリリース, “効率的なネットワーク構成を示すグラフ発見を競うコンペを開催/スパコン内のCPU、あなたならどう接続しますか?,” <https://www.nii.ac.jp/news/release/2017/0306.html>, March 2017.
- [9] 国立情報学研究所ニュースリリース, “効率的なネットワーク構成を示すグラフ発見を競うコンペを今年も開催/スパコン内のCPU、あなたならどう接続しますか?,” <https://www.nii.ac.jp/news/release/2018/0516.html>, May 2018.
- [10] 国立情報学研究所ニュースリリース, “100万個のCPUを効率的に接続するグラフの発見者求む!～未来スパコンのネットワーク構成を発見するコンペ「グラフ ゴルフ 2019」を今年も開催～,” <https://www.nii.ac.jp/news/release/2019/0426.html>, April 2019.
- [11] 日刊工業新聞 BusinessLine, “国立情報学研究所、単純構成のグラフ競うコンペの参加者募集、効率的なスパコン設計へ,” June 2015.
- [12] I. Watch, “スパコン内CPUコアの効率的な接続へ、より単純なネットワークポロジータを競うコンペ「グラフゴルフ」、NIIが開催,” <http://internet.watch.impress.co.jp/docs/news/1048249.html>, March 2017.
- [13] 科学新聞 2面, “次世代のスパコン設計を模した40万頂点数の巨大グラフを発見～通信遅延の大幅な低下などの実用に期待～効率的なスパコン設計につながるグラフ発見を競うコンペ「グラフゴルフ」で～,” <http://internet.watch.impress.co.jp/docs/news/1048249.html>, Dec. 2018.
- [14] 藤原一毅, 鯉渕道紘, 藤田聡, 中野浩嗣, “開催報告 (Graph Golf 2015),” 電子情報通信学会情報・システムソサイエティ誌, p.26, 2016.
- [15] 藤原一毅, 鯉渕道紘, 藤田聡, 中野浩嗣, “開催報告 (Graph Golf 2016),” 電子情報通信学会情報・システムソサイエティ誌, p.14, 2017.
- [16] 藤原一毅, 鯉渕道紘, 藤田聡, 中野浩嗣, “開催報告 (Graph Golf 2017),” 電子情報通信学会情報・システムソサイエティ誌, p.7, 2018.
- [17] 藤原一毅, 鯉渕道紘, 藤田聡, 中野浩嗣, “開催報告 (Graph Golf 2018),” 電子情報通信学会情報・システムソサイエティ誌, p.14, 2019.
- [18] 国立情報学研究所ニュースリリース, “優れたグラフ発見の3チーム計7名を表彰/応募は284件～効率的なスパコン設計につながるグラフ発見を競うコンペ「グラフゴルフ」～,” <https://www.nii.ac.jp/news/release/2015/1210.html>, Dec. 2015.
- [19] 国立情報学研究所ニュースリリース, “優れたグラフ発見の3チームと個人1名を表彰/～効率的なスパコン設計につながるグラフ発見を競うコンペ「グラフゴルフ」～,” <https://www.nii.ac.jp/news/release/2016/1122.html>, Nov. 2016.
- [20] 国立情報学研究所ニュースリリース, “優れたグラフを発見した応募者を表彰/効率的なスパコン設計につながるグラフ発見を競うコンペ「グラフゴルフ」,” <https://www.nii.ac.jp/news/release/2017/1122.html>, Nov. 2017.
- [21] 国立情報学研究所ニュースリリース, “次世代のスパコン設計を模した40万頂点数の巨大グラフを発見～通信遅延の大幅な低下などの実用に期待～効率的なスパコン設計につながるグラフ発見を競うコンペ「グラフゴルフ」で～,” <https://www.nii.ac.jp/news/release/2018/1127.html>, Nov. 2018.
- [22] 藤原一毅, 藤田聡, 中野浩嗣, 井上武, 鯉渕道紘, “みんながOrder/Degree問題を解いて究極の低遅延相互結合網をつくらう,” 電子情報通信学会 コンピュータシステム研究会, 電子情報通信学会技術研究報告, vol.115, no.174, pp.223–228, Aug. 2015.
- [23] 石川裕, 他, “HPCI技術ロードマップ白書,” March 2012.
- [24] N. Shimizu and R. Mori, “Average shortest path length of graphs of diameter 3,” 2016 Tenth IEEE/ACM International Symposium on Networks-on-Chip (NOCS), pp.1–6, Aug. 2016.
- [25] M. Nakao, H. Murai, and M. Sato, “A Method for Order/Degree Problem Based on Graph Symmetry and Simulated Annealing with MPI/OpenMP Parallelization,” Proc. International Conference on High Performance Computing in Asia-Pacific Region, pp.128–137, HPC Asia 2019, 2019.
- [26] R. Mizuno and Y. Ishida, “Constructing large-scale low-latency network from small optimal networks,” 2016 Tenth IEEE/ACM International Symposium on Networks-on-Chip (NOCS), pp.1–5, Aug. 2016.
- [27] T. Kitasuka and M. Iida, “A heuristic method of generating diameter 3 graphs for order/degree problem (invited paper),” 2016 Tenth IEEE/ACM International Symposium on Networks-on-Chip (NOCS), pp.1–6, Aug. 2016.
- [28] R. Yasudo, H. Matsutani, M. Koibuchi, H. Amano, and T. Nakamura, “Scalable Networks-on-Chip with Elastic Links Demarcated by Decentralized Routers,” IEEE Trans. Computers, vol.66, no.4, pp.702–716, 2017.
- [29] K. Nakano, D. Takafuji, S. Fujita, H. Matsutani, I. Fujiwara, and M. Koibuchi, “Randomly Optimized Grid Graph for Low-Latency Interconnection Networks,” 45th International Conference on Parallel Processing (ICPP), pp.340–349, 2016.
- [30] A. Singla, C.-Y. Hong, L. Popa, and P.B. Godfrey, “Jellyfish: networking data centers randomly,” USENIX Conference on Networked Systems Design and Implementation (NSDI), p.17, April 2012.
- [31] M. Koibuchi, H. Matsutani, H. Amano, D.F. Hsu, and H. Casanova, “A case for random shortcut topologies for HPC interconnects,” International Symposium on Computer Architecture (ISCA), pp.177–188, June 2012.
- [32] M. Ogleary, Y. Yu, C. Qian, E.L. Miller, and J. Zhao, “String figure: A scalable and elastic memory network architecture,” 25th IEEE International Symposium on High Performance Computer Architecture (HPCA), pp.647–660, 2019.
- [33] U.Y. Ogras and R. Marculescu, “It’s a small world after all”: NoC performance optimization via long-range link insertion,” IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol.14, no.7, pp.693–706, 2006.

# 付 録

表 A-1: 一般グラフ部門の出題と発見された最善解の一覧

頂点数	次数	直径			平均距離			年
		ランダム	最善解	下界	ランダム	最善解	下界	
16	3	5	3	3	2.450000	2.200000	2.200000	2015
16	4	3	3	2	1.916667	1.750000	1.733333	2015
32	5	4	3	3	2.280242	2.032258	2.032258	2017
36	3	6	4	4	3.457143	3.066667	3.057143	2016
64	3	8	5	5	4.303075	3.769841	3.761905	2015
64	4	5	4	4	3.154266	2.869048	2.857143	2015
64	8	3	3	2	2.170635	1.926587	1.873016	2016
64	16	3	2	2	1.753968	1.746032	1.746032	2015
72	4	6	4	4	3.215962	2.985915	2.985915	2018
96	3	9	6	6	4.849781	4.272368	4.200000	2016
256	3	11	8	7	6.125460	5.636366	5.588235	2015
256	4	7	6	5	4.416391	4.133762	4.090196	2015
256	5	6	5	4	3.732445	3.493137	3.470588	2018
256	8	4	4	3	2.888143	2.735968	2.717647	2016
256	10	4	3	3	2.648713	2.568627	2.568627	2018
256	16	3	3	2	2.281863	1.991728	1.937255	2016
256	18	3	3	2	2.188082	1.938603	1.929412	2017
256	23	3	2	2	2.011581	1.909804	1.909804	2015
300	7	5	4	3	3.172642	3.012285	2.812709	2016
384	3	11	9	8	6.648961	6.225617	6.065274	2016
512	8	5	4	4	3.267934	3.111378	2.966732	2016
576	30	3	2	2	2.141057	1.947826	1.947826	2017
1024	3	13	11	9	8.067361	7.678970	7.527859	2016
1024	8	5	5	4	3.611034	3.499511	3.483871	2017
1024	11	5	4	3	3.174107	3.053763	2.870968	2017
1024	32	3	3	2	2.324601	1.997844	1.968719	2016
1344	30	3	3	3	2.482569	2.345821	2.307520	2017
1560	40	3	3	2	2.323343	2.038230	1.974343	2016
1800	7	6	5	4	4.179384	4.076987	3.801556	2017
2300	10	5	5	4	3.652002	3.587647	3.556329	2018
3019	30	4	3	3	2.730300	2.693229	2.691849	2018
3250	57	3	3	2	2.343608	2.070020	1.982456	2016
4096	3	15	13	11	10.100137	9.786842	9.508425	2015
4096	4	10	9	7	6.915910	6.752513	6.469597	2015
4096	16	5	4	4	3.323065	3.252718	2.991941	2015
4096	23	4	4	3	2.929344	2.886137	2.865201	2015
4096	60	3	3	3	2.395112	2.295216	2.106227	2016
4096	64	3	3	2	2.346357	1.999504	1.984371	2016
4855	30	4	4	3	2.827926	2.808886	2.808405	2018
4896	24	4	4	3	2.948959	2.898227	2.877426	2017
9344	10	6	5	5	4.311007	4.246541	4.013165	2017
10000	3	16	15	12	11.373544	11.111414	10.774977	2015
10000	4	11	10	8	7.727035	7.601184	7.345535	2015
10000	7	7	6	5	5.136482	5.072707	4.782978	2017
10000	11	6	5	4	4.165354	4.106411	3.864686	2017
10000	16	5	5	4	3.655512	3.625174	3.587159	2015
10000	20	5	4	4	3.424385	3.375238	3.195920	2017
10000	23	4	4	3	3.256133	3.200257	2.944794	2015
10000	60	3	3	3	2.690121	2.648977	2.633963	2016
10000	64	3	3	3	2.655734	2.609927	2.583958	2015



12000	7	7	7	6	5.237548	5.176015	4.911993	2018
20000	11	6	6	5	4.478759	4.443895	4.321266	2018
40000	8	7	7	6	5.500540	5.465012	5.346584	2018
77000	6	9	9	7	6.882292	6.834647	6.619657	2018
88128	12	7	6	5	4.910985	4.882785	4.780771	2017
98304	10	7	7	6	5.382717	5.355208	5.155367	2017
100000	7	9	8	7	6.415420	6.387834	6.216272	2017
100000	11	7	6	5	5.165934	5.143691	4.864259	2017
100000	20	5	5	4	4.157337	4.132631	3.919599	2017
100000	32	5	4	4	3.717217	3.705557	3.671677	2017
100000	64	4	4	3	3.034717	3.015370	2.958400	2017
132000	8	8	8	6	6.120818	6.094654	5.801998	2018
200000	32	5	5	4	3.855047	3.849094	3.835839	2018
200000	64	4	4	3	3.254346	3.236273	2.979200	2018
400000	32	5	5	4	4.006156	3.996818	3.917920	2018

表 A・2: 格子グラフ部門の出題と発見された最善解の一覧

頂点数	次数	辺長制約	直径			平均距離			年
			ランダム	最善解	下界	ランダム	最善解	下界	
4 × 4	3	2	9	3	3	3.833333	2.200000	2.200000	2017
4 × 16	4	4	8	5	4	3.939980	2.969742	2.857143	2018
16 × 16	3	3	22	10	7	8.584252	5.920159	5.588235	2017
16 × 16	3	4	18	9	7	7.890993	5.753002	5.588235	2017
16 × 16	3	15	13	8	7	6.274295	5.667249	5.588235	2017
16 × 16	6	3	13	10	4	5.322304	4.155699	3.105882	2017
16 × 16	6	4	10	8	4	4.552083	3.581005	3.105882	2017
16 × 16	6	15	5	4	4	3.397273	3.190686	3.105882	2017
16 × 16	15	3	10	10	3	4.216513	3.913358	2.058824	2017
16 × 16	15	4	8	8	3	3.476930	3.121967	2.058824	2017
16 × 16	15	15	4	3	3	2.376379	2.179289	2.058824	2017
32 × 32	4	3	28	21	6	11.649543	7.963815	5.300098	2018
32 × 32	4	4	21	16	6	9.303009	6.658413	5.300098	2018
32 × 32	4	5	19	13	6	8.403203	6.062422	5.300098	2018
16 × 64	4	4	32	20	6	12.277917	7.726016	5.300098	2018
16 × 64	4	5	26	16	6	10.398900	6.757948	5.300098	2018
16 × 64	4	7	21	12	6	8.458341	5.950437	5.300098	2018
4 × 256	4	12	37	22	6	13.378244	8.452216	5.300098	2018
4 × 256	4	18	24	15	6	9.525652	6.632465	5.300098	2018
4 × 256	4	24	19	11	6	8.020914	6.003484	5.300098	2018
100 × 100	3	6	59	33	12	23.753370	14.847902	10.774977	2017
100 × 100	3	18	26	17	12	13.147303	11.432421	10.774977	2017
100 × 100	3	33	19	16	12	11.676544	11.249062	10.774977	2017
100 × 100	9	6	36	33	5	13.570290	11.589964	4.398740	2017
100 × 100	9	18	14	11	5	6.175121	5.151371	4.398740	2017
100 × 100	9	33	9	7	5	4.923812	4.560467	4.398740	2017
100 × 100	28	6	34	33	3	11.826052	11.533493	2.918792	2017
100 × 100	28	18	12	11	3	4.821576	4.394069	2.918792	2017
100 × 100	28	33	7	6	3	3.533770	3.258053	2.918792	2017

ここに示した下界は Moore bound に基づく値である。