MPI/OpenMP並列によるグラフ対称性とSimulated Annealingを用いたOrder/Degree問題の一解法

中尾 昌広，村井 均，佐藤三久（理化学研究所 計算科学研究センター）
Background (1/3)

Order/Degree Problem (ODP) in graph theory
- Find a graph with smallest diameter and average shortest path length (ASPL) given an order (n: 頂点数) and a degree (d: 次数)
- Degree is a number of edges connected to one node
- ASPL is an average of the distances between all nodes
- Examples of the graph with (n, d) = (10, 3)

Diameter = 3 (maximum value in distances)
ASPL = 85/45 = 1.88...

distance matrix
Background (1/3)

- **Order/Degree Problem (ODP) in graph theory**
  - Find a graph with smallest **diameter** and **average shortest path length (ASPL)** given an order \((n: \text{頂点数})\) and a degree \((d: \text{次数})\)
  - Degree is a number of edges connected to one node
  - ASPL is an average of the distances between all nodes
  - Examples of the graph with \((n, d) = (10, 3)\)

Need 3 hops

Distance matrix:
- Diameter = 3 (maximum value in distances)
- ASPL = \(85/45 = 1.88...\)
Background (1/3)

Order/Degree Problem (ODP) in graph theory
- Find a graph with smallest diameter and average shortest path length (ASPL) given an order (n: 頂点数) and a degree (d: 次数)
- Degree is a number of edges connected to one node
- ASPL is an average of the distances between all nodes
- Examples of the graph with (n, d) = (10, 3)

Diameter = 3 (maximum value in distances)
ASPL = 85/45 = 1.88...
Background (1/3)

- **Order/Degree Problem (ODP) in graph theory**
  - Find a graph with smallest **diameter** and **average shortest path length (ASPL)** given an order \(n\) and a degree \(d\)
  - Degree is a number of edges connected to one node
  - ASPL is an average of the distances between all nodes
  - Examples of the graph with \((n, d) = (10, 3)\)
Background (2/3)

- ODP is useful for design of network topology with low latency
  - Network topology in HPC systems affects its performance
  - Supercomputers

- Data centers in Google, Amazon etc.
- Processor chip (e.g. Knights Landing has 72 cores)
  - Packet network in many core processor

https://www.extremetech.com/extreme/171678-intel-unveils-72-core-x86-knights-landing-cpu-for-exascale-supercomputing
Background (3/3)

- **Graph Golf Competition for ODP**
  - [http://research.nii.ac.jp/graphgolf](http://research.nii.ac.jp/graphgolf)
  - Held by National Institute of Informatics, Japan since 2015
  - **Rule**
    - Find each graph with the smallest **diameter** and **ASPL** in 14 problems in 2018
    - Problems are changed every year
    - Participants uploads their graphs on the official website within a limited period (May 14 to Oct. 14, 2018)

<table>
<thead>
<tr>
<th>No</th>
<th>Problem (n, d)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>72, 4</td>
</tr>
<tr>
<td>2</td>
<td>256, 5</td>
</tr>
<tr>
<td>3</td>
<td>256, 10</td>
</tr>
<tr>
<td>4</td>
<td>2300, 10</td>
</tr>
<tr>
<td>5</td>
<td>3019, 30</td>
</tr>
<tr>
<td>6</td>
<td>4855, 30</td>
</tr>
<tr>
<td>7</td>
<td>12000, 7</td>
</tr>
<tr>
<td>8</td>
<td>20000, 11</td>
</tr>
<tr>
<td>9</td>
<td>40000, 8</td>
</tr>
<tr>
<td>10</td>
<td>77000, 6</td>
</tr>
<tr>
<td>11</td>
<td>132000, 8</td>
</tr>
<tr>
<td>12</td>
<td>200000, 32</td>
</tr>
<tr>
<td>13</td>
<td>200000, 64</td>
</tr>
<tr>
<td>14</td>
<td>400000, 32</td>
</tr>
</tbody>
</table>
Objective

- Proposes an algorithm for efficiently solving ODP
  - Based on "graph symmetry" and "simulated annealing (SA)"
  - Evaluate "search performance" and "speed performance"
  - Won award of Graph Golf 2018

Award ceremony, CANDAR'18, Takayama, Gifu, Japan
Agenda

- Background
- Related works
- Explain our algorithm
- Evaluation of search and speed performances
- Summary
Related Works (2/3)

- **Randomized topology** [Koibuchi2012, Koibuchi2013, Singla2012]
  - When connecting nodes randomly, the distance shrinks due to the **small world effect**, and its latency decreases
  - Extensibility and failure resistance
  - For cases where the cable length becomes long, it can be relaxed by rack placement optimization

![Diagram: Regular Topology](https://www.nii.ac.jp/userimg/openhouse/2015/B11.pdf)

- Regular Topology
  - (It takes a long time for the message to arrive)

- Randomized Topology
  - (Shortcut allows a message to arrive soon)

Related Works (3/3)

- Are randomized topologies best? → No

From the numbers of nodes $n$ and degrees $d$, the theoretical diameter lower bound $K_{\{n,d\}}$ and the ASPL lower bound $L_{\{n,d\}}$ are calculated.

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

$$L_{n,d} = \begin{cases} 1 & \text{if } K_{n,d} = 1 \\ \frac{1}{S_{n,d} + K_{n,d}R_{n,d}} & \text{if } K_{n,d} \geq 2 \end{cases}$$

$$S_{n,d} = \sum_{i=1}^{K_{n,d}-1} id(d-1)^{i-1}$$

$$R_{n,d} = n - 1 - \sum_{i=1}^{K_{n,d}-1} d(d-1)^{i-1}$$

Difference between ASPL and theoretical lower bound is large

Agenda

- Background
- Related works
- **Explain our algorithm**
- Evaluation of search and speed performances
- Summary
What are difficult points in ODP?

(1) The number of graphs satisfying the given number of nodes and degrees is enormous
   - Number of graphs satisfying \((n, d) = (12, 4)\) is about 480 Billions
   - It is difficult to find the best solution because the problem has many local optima

(2) The calculation time required for obtaining ASPL is enormous
   - The calculation complexity with \(n\) nodes and \(d\) degrees is \(O(n^2d)\)
   - For the graph with \((n, d) = (400000, 32)\), the calculation time required for obtaining ASPL is about 5.5 hours on Xeon Ivy Bridge
     (Diameter can be calculated simultaneously with ASPL)
Origin of idea

- What kind of topology does small diameter and ASPL have?
  - Randomized topology is better than regular topology for latency
  - But, randomized topology is not best

Regular Topology
(It takes a long time for the message to arrive)

Randomized Topology
(Shortcut allows a message to arrive soon)
Origin of idea

- What kind of topology does small diameter and ASPL have?
  - Randomized topology is better than regular topology for latency
  - But, randomized topology is not best

Topology with both regular and randomized properties may be better
Approach

(1) Make the network topology **symmetrical**
- SA's search performance may be improved due to reduce search space
- The reason for using SA is that it doesn't easily fall into local optima

(2) Reducing the calculation time of ASPL
- Reduce calculation time of ASPL by the network topology symmetrical (Up to 10,000 times)
- Hybrid parallelization with MPI and OpenMP on our cluster system (Up to 200 times on 400 CPU cores)
- Calculation time of ASPL decreased from 5.5 hours to 0.01 sec.

Our source code is available at
[https://github.com/mnakao/GraphGolf](https://github.com/mnakao/GraphGolf)

about 2,000,000 times faster
(= 10,000 x 200)
Graph symmetry

- Examples of the graph symmetry with \((n, d) = (24, 3)\)

The variable \(g\) is the number of groups (\(g\) must be a divisor of \(n\))
- When a graph is viewed as a plane, if it is rotated by \(360/g\) degrees, the connection relationship between the edge and the node becomes the same graph
- For the case of \(g = 1\), a normal graph (not symmetrical) is obtained
SA with Graph symmetry

(1) Create a random graph with the number of nodes of the target graph divided by g, and duplicate g the graphs (the graph with \( (n, d) = (6, 3) \) is created \( \times 4 \)). And select one edge from each graph.

(2) Connect both sides so that it becomes symmetrical.

- Initialize a solution \( (n, d, g) = (24, 3, 4) \)
SA with Graph symmetry

- Mutate a new solution \((n, d, g) = (24, 3, 2)\)

1. Randomly select two edges from all the edges
2. Select edges symmetrically related to (1)
3. Apply the 2-opt method to the edges selected in (1)
4. Apply the 2-opt method to (2) in the same way as (3)
SA with Graph symmetry

- Reduce the calculation time for ASPL
  - In general, it calculates the distance from all nodes to all other nodes using BFS
  - However, with graph symmetry, the distances from the node to all other nodes are the same for all symmetrically related nodes
  - Thus, the complexity becomes \(O(n^2 \cdot d/g)\) from \(O(n^2 \cdot d)\)
Agenda

- Background
- Related works
- Explain our algorithm
- Evaluation of search and speed performances
- Summary
Search Performance

- In vertical axis, **ASPL Gap** is a difference from theoretical lower bound
- Proposed method is executed 100 times with different $g$
- Search performance tends to increase as the $g$ increases
- However, the problem $\text{g} = (72, 5)$, the search performance is better for $g=9$ than for $g=12$, indicating that the solution performance may deteriorate if the value of $g$ is too large
  - The $g$ expresses the strength of regularity of a graph; regularity becomes stronger as $g$ increases
Search Performance

(n, d) = (72, 5)

(n, d) = (256, 5)

(n, d) = (256, 10)

Theoretical lower bound

Best solution in Graph Golf

Theoretical lower bound
How do I set the value of $g$?

- Firstly, set a value of $g$ which is as large as possible
- Next, gradually reduce the value of $g$
- $(n, d) = (132000, 8)$
Speed Performance

- COMA cluster system at University of Tsukuba

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>Intel Xeon-E5 2670v2 2.8 GHz x 2 Sockets</td>
</tr>
<tr>
<td>Memory</td>
<td>DDR3 1866MHz 59.7GB/s 64GB</td>
</tr>
<tr>
<td>Network</td>
<td>InfiniBand FDR 7GB/s</td>
</tr>
<tr>
<td>Software</td>
<td>intel/16.0.2, intelmpi/5.1.1, Omni Compiler 1.2.1</td>
</tr>
<tr>
<td></td>
<td>Python 2.7.9, networkx 1.9</td>
</tr>
</tbody>
</table>

- Speed-up techniques
  - Graph symmetry
  - Hybrid parallelization with MPI and OpenMP
Performance results by Graph Symmetry

- Measure time to calculate ASPL 100 times

![Graphs showing time vs. number of groups for different (n, d, g) values]

- The bar graph shows the time on the left vertical axis, and the line graph shows the speed up ratio with $g = 1$ on the right vertical axis.

- Speed ups of $8.11$, $31.76$, $15.67$ times, respectively, were achieved for $(n, d, g) = (72, 4, 12)$, $(256, 5, 32)$, and $(256, 10, 16)$. 
Performance results by Hybrid Parallelization

- Multiple BFS are simultaneously executed using MPI, and each BFS is executed in parallel using several OpenMP threads.
- Number of Process is $P$, and Number of Threads is $T$.
- The maximum value $P$ is $(n/g)$.
- In problem $(n, d, g) = (400000, 32, 10000)$, the performance obtained with 40P10T is 209.80 (5.97 x 35.11) times better than that obtained with 1P1T.

Thread parallelization with OpenMP

Hybrid parallelization with MPI and OpenMP
# Results of Graph Golf 2018

<table>
<thead>
<tr>
<th>No</th>
<th>Problem (n, d)</th>
<th>Groups</th>
<th>ASPL Gap</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>72, 4</td>
<td>9</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>256, 5</td>
<td>32</td>
<td>0.02255</td>
</tr>
<tr>
<td>3</td>
<td>256, 10</td>
<td>16</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>2300, 10</td>
<td>115</td>
<td>0.03132</td>
</tr>
<tr>
<td>5</td>
<td>3019, 30</td>
<td>15</td>
<td>0.00237</td>
</tr>
<tr>
<td>6</td>
<td>4855, 30</td>
<td>15</td>
<td>0.00057</td>
</tr>
<tr>
<td>7</td>
<td>12000, 7</td>
<td>1000</td>
<td>0.26531</td>
</tr>
<tr>
<td>8</td>
<td>20000, 11</td>
<td>1000</td>
<td>0.12263</td>
</tr>
<tr>
<td>9</td>
<td>40000, 8</td>
<td>1600</td>
<td>0.12066</td>
</tr>
<tr>
<td>10</td>
<td>77000, 6</td>
<td>2200</td>
<td>0.22312</td>
</tr>
<tr>
<td>11</td>
<td>132000, 8</td>
<td>4400</td>
<td>0.29266</td>
</tr>
<tr>
<td>12</td>
<td>200000, 32</td>
<td>5000</td>
<td>0.01362</td>
</tr>
<tr>
<td>13</td>
<td>200000, 64</td>
<td>2500</td>
<td>0.25707</td>
</tr>
<tr>
<td>14</td>
<td>400000, 32</td>
<td>10000</td>
<td>0.07890</td>
</tr>
</tbody>
</table>

Awarded: Proposed method won 8 problems in 14 problems.
Haruishi's algorithm uses **voltage graphs**

- **Regular topology**
- **Merit**
  - Relatively small diameter and ASPL
  - Calculation time for ASPL using regularity is small
- **Demerit**
  - Diameter and ASPL may not be small in some cases
  - Specific number of nodes

Agenda

- Background
- Related works
- Explain our algorithm
- Evaluation of search and speed performances

Summary
In the proposed method, the topology is made symmetrical, making it possible to efficiently find a good solution. Making the topology of the graph symmetrical reduced the calculation time required for the ASPL. Moreover, by utilizing hybrid parallelization with MPI and OpenMP, the calculation time for the ASPL was further reduced. A performance improvement of 209.80 times was achieved for the problem \((n, d, g) = (400000, 32, 10000)\) using only the hybrid parallelization. In addition, since graph symmetry was also applied, the performance improvement was about 2,098,000 times compared to that obtained with \(g=1\). The calculation time of ASPL decreased from 5.5 hours to 0.01 seconds.
Future works

- Graph Symmetry is basic idea.
- It can be combined with optimization algorithm other than SA
- e.g. Deep Learning, Genetic Algorithms, etc.
When $n$ is a prime number

- In Graph Golf 2018, there is a problem with $(n, d) = (3019, 30)$
- When $n$ is a prime number, $g$ cannot be set in the method explained so far
  - Extend the method to deal with cases where $n$ is a prime number
  - Add **center points** to the graph

In addition, the following combinations are possible.

- $n_{503d30} \times g_{6} + c_{1} = n_{3019d30}$
  - groups = 6  centers = 1
  - $(n, d) = (503, 30)$  $503 \times 6 + 1 = 3019$

- $n_{301d30} \times g_{10} + c_{9} = n_{3019d30}$
- $n_{200d30} \times g_{15} + c_{19} = n_{3019d30}$
- $n_{100d30} \times g_{30} + c_{19} = n_{3019d30}$
How are parameters of SA set?

- 最高温度：初期解（ランダムグラフ）に対して100回ランダムにエッジを交換して得られるエネルギーの差を50%で受理

- 最低温度：理論的な最小のエネルギー差（2/(n*(n-1))）を0.01%で受理

- クーリング率：1

- 総計算回数：100万回

詳細は予稿集の4.1「パラメータの設計指針」を参照ください