A Method for Order/Degree Problem Based on Graph Symmetry and Simulated Annealing

Masahiro Nakao (RIKEN Center for Computational Science)
What is Order/Degree Problem (ODP)?

- Find the graph with the smallest **diameter** and **average shortest path length (ASPL)**
- Given order \(n\) and degree \(d\) pairs
  - Examples of the graph with \((n, d) = (10, 3)\)

Diameter=3, ASPL=1.89 (Random)  
Diameter=2, ASPL=1.67 (Optimal)
What are difficult points in ODP?

(1) The number of graphs satisfying the given number of vertices and degrees is enormous
- 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 \( n \) vertices and \( d \) degrees is \( O(n^{2d}) \)
- For the graph with \( (n, d) = (400,000, 32) \), the calculation time required for obtaining ASPL is about 5.5 hours on Xeon Ivy Bridge
Approach

- Make the network topology *symmetrical*, thereby
  1. Improving the solution search performance of simulated annealing (SA)
  2. Reducing the calculation time of ASPL

- Hybrid parallelization with MPI and OpenMP is applied to further reduce the calculation time on our cluster system

  The calculation time of ASPL decreased from 5.5 hours to **0.01 seconds**

  about 2,000,000 times faster
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 vertex becomes the same graph
- For the case of \(g = 1\), a normal graph (not symmetrical) is obtained
SA with Graph symmetry

1. Set initialize parameters
   - $k \leftarrow 1$
   - Initial temperature: $T_i$
   - Initial solution: $x_i$
   - Initial energy: $E(x_i)$

2. Generate next solution
   - $x \leftarrow x_i$

3. Compute energy
   - $\Delta E = E(x') - E(x_i)$

4. Acceptance?
   - Yes

5. Transition
   - $x_{i+1} \leftarrow x'$, $E(x_{i+1}) \leftarrow E(x')$

6. Cooling Cycle?
   - No

7. Cooling
   - $T_{i+1} \leftarrow T_i$

8. Terminal?
   - Yes

Generate initial solution $(n, d, g) = (24, 3, 4)$

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

2. Connect both sides so that it becomes symmetrical
SA with Graph symmetry

- Generate 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

- Reduction the calculation time for ASPL
  - In general, it is necessary to calculate the distance from all vertices to all other vertices using BFS
  - However, with graph symmetry, the distances from the vertex to all other vertices are the same for all symmetrically related vertices
  - Thus, the complexity becomes $O(n^2d/g)$ from $O(n^2d)$

```
(1) Set initialize parameters
   k ← 1
   Initial temperature: $T_i$
   Initial solution: $x_i$
   Initial energy: $E(x_i)$

(2) Generate next solution
   $x ← x_2$

(3) Compute energy
   $\Delta E = E(x') - E(x)$

(4) Acceptance?
   No

(5) Transition
   $x_{i+1} ← x'$, $E(x_{i+1}) ← E(x)$

(6) Cooling Cycle?
   No

(7) Cooling
   $T_{i+1} ← T_i$

(8) Terminal?
   Yes
```

$g=1$  $g=2$
Search Performance

- Proposed method is executed 100 times with different g
- The solution search performance tends to increase as the g increases
- However, the problem \((n, d) = (72, 5)\), the solution search performance is better for \(g=9\) than for \(g=12\), indicating that solution search 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

\[(n, d) = (72, 5)\]  \[(n, d) = (256, 5)\]  \[(n, d) = (256, 10)\]
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 $g$
- $(n, d) = (132000, 8)$
When \( n \) is a prime number

- In GraphGolf 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

\[
\begin{align*}
n503d30 \times g6 + c1 &= n3019d30 \\
groups &= 6 \quad centers = 1
\end{align*}
\]

\((n, d) = (503, 30)\)

\[503 \times 6 + 1 = 3019\]

In addition, the following combinations are possible.

\[
\begin{align*}
n301d30 \times g10 + c9 &= n3019d30 \\
n200d30 \times g15 + c19 &= n3019d30 \\
n100d30 \times g30 + c19 &= n3019d30
\end{align*}
\]
Speed Performance

- COMA cluster system at University of Tsukuba

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></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

- The COMA system provided by Interdisciplinary Computational Science Program in the Center for Computational Sciences, University of Tsukuba
  - Computing resources such as COMA and Oakforest-PACS can be used for free

- Entries are held every December
Performance results by Graph Symmetry

- Measure time to calculate ASPL 100 times

\[(n, d) = (72, 5) \quad x \quad 8.11\]
\[(n, d) = (256, 5) \quad x \quad 31.76\]
\[(n, d) = (256, 10) \quad x \quad 15.67\]

- 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.
- The calculation complexity for the ASPL becomes $O(n^2d/(g*P*T))$ from $O(n^2d/g)$ when the number of MPI ranks is $P$ and the number of threads is $T$.
- The largest problem $(n, d, g) = (400000, 32, 10000)$ in Graph Golf 2018 is used.
- The performance obtained with 40P10T is 209.80 times higher than that obtained with 1P1T.

Thread parallelization with OpenMP

Hybrid parallelization with MPI and OpenMP
## Results

<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>4000000, 32</td>
<td>10000</td>
<td>0.07890</td>
</tr>
</tbody>
</table>

Proposed method won 8 problems in 14 problems
For more information

- Source Code
  - https://github.com/mnakao/GraphGolf

- Publication
  - 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, 第167回HPC研究会, 沖縄, 2018年12月
  - A Method for Order/Degree Problem Based on Graph Symmetry and Simulated Annealing with MPI/OpenMP Parallelization, HPC Asia 2019, Guangzhou, China, Jan. 2019
Conclusion

- 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.