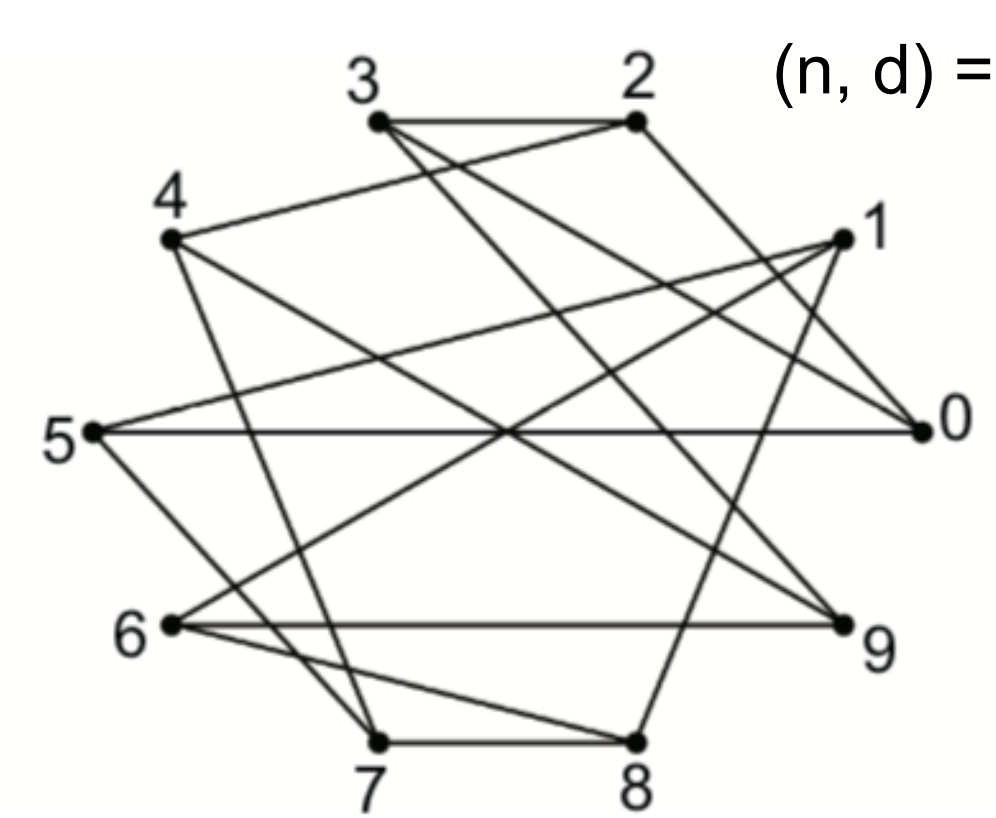
A Method for Order/Degree Problem based on Graph Symmetry and Simulated Annealing with MPI/OpenMP Parallelization

Masahiro Nakao, Hitoshi Murai, Mitsuhisa Sato

(RIKEN Center for Computational Science)

Background (1/3)

- **Order/Degree Problem (ODP) in graph theory**
  - Find a graph with the smallest **diameter** and **average shortest path length (ASPL)** given an order n and a degree d
  - diameter is a maximum distance between nodes
  - ASPL is an average distance between all nodes
  - Degree is a number of edges connected to one node



(n, d) = (10, 3)

	0	1	2	3	4	5	6	7	8	9
0		2	1	1	2	1	3	2	3	2
1			3	3	3	1	1	2	1	2
2				1	1	2	3	2	3	2
3					2	2	2	3	3	1
4						2	2	1	2	1
5							2	1	2	3
6								2	1	1
7									1	2
8										2
9										

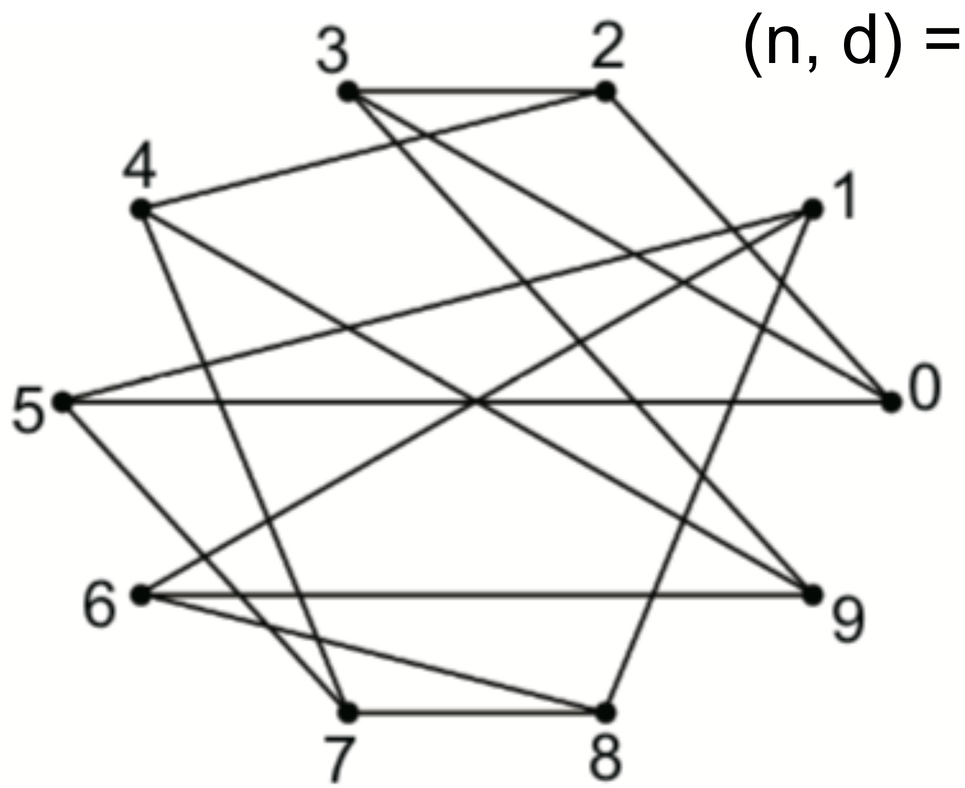
distance matrix

Diameter = 3

ASPL = 85/45
= 1.88...

Background (1/3)

- **Order/Degree Problem (ODP) in graph theory**
  - Find a graph with the smallest **diameter** and **average shortest path length (ASPL)** given an order n and a degree d
  - diameter is a maximum distance between nodes
  - ASPL is an average distance between all nodes
  - Degree is a number of edges connected to one node



(n, d) = (10, 3)

	0	1	2	3	4	5	6	7	8	9
0		2	1	1	2	1	3	2	3	2
1			3	3	3	1	1	2	1	2
2				1	1	2	3	2	3	2
3					2	2	2	3	3	1
4						2	2	1	2	1
5							2	1	2	3
6								2	1	1
7									1	2
8										2
9										

distance matrix

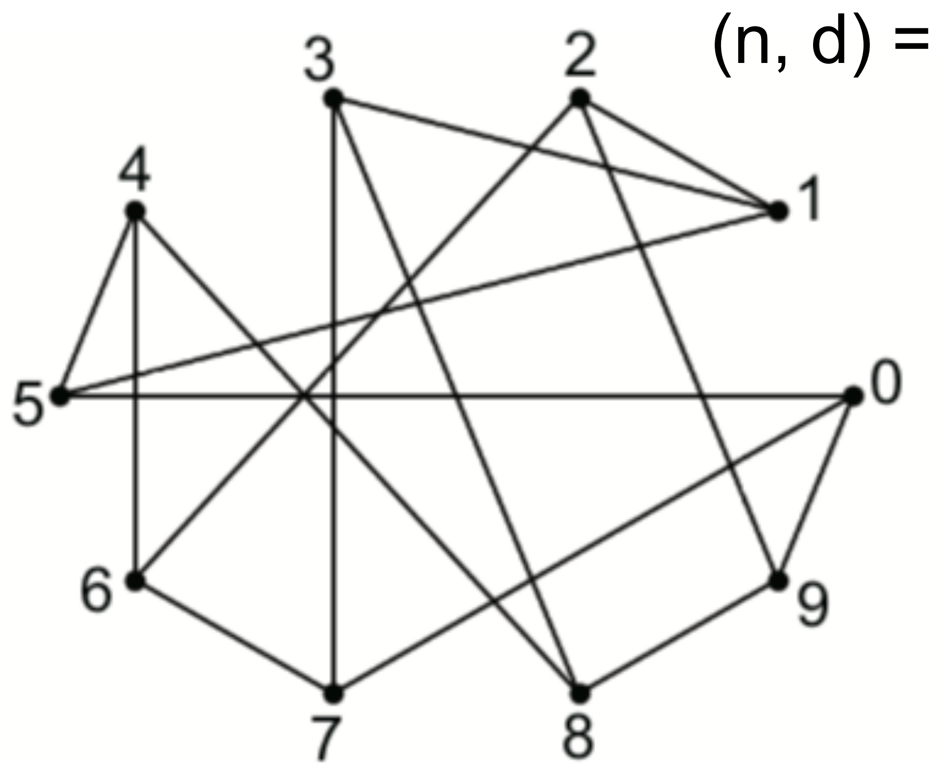
Diameter = $\boxed{3}$

ASPL = 85/45
= 1.88...

Background (1/3)

- **Order/Degree Problem (ODP) in graph theory**
  - Find a graph with the smallest **diameter** and **average shortest path length (ASPL)** given an order n and a degree d
  - diameter is a maximum distance between nodes
  - ASPL is an average distance between all nodes
  - Degree is a number of edges connected to one node

(n, d) = (10, 3)



	0	1	2	3	4	5	6	7	8	9
0		2	2	2	2	1	2	1	2	1
1			1	1	2	1	2	2	2	2
2				2	2	2	1	2	2	1
3					2	2	2	1	1	2
4						1	1	2	1	2
5							2	2	2	2
6								1	2	2
7									2	2
8										1
9										

distance matrix

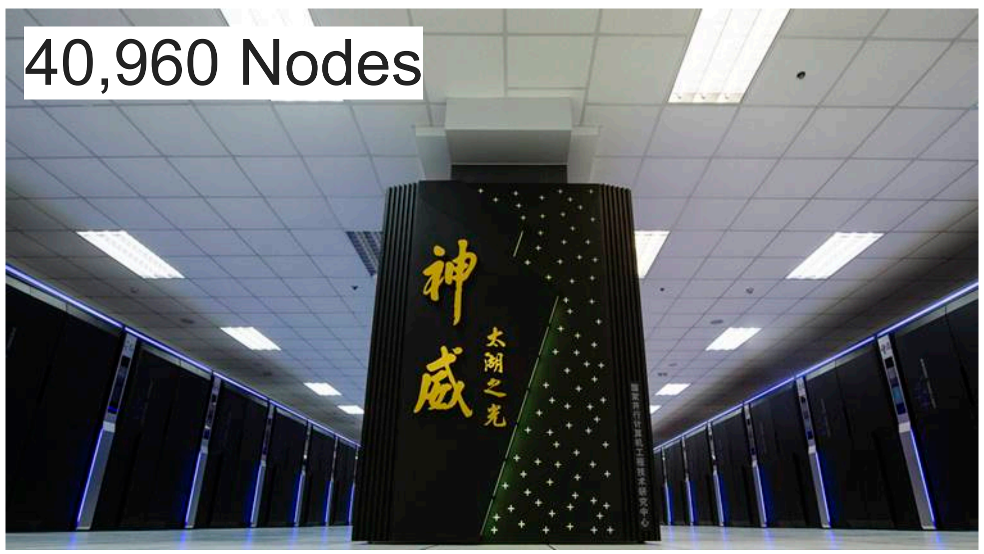
Diameter = $\boxed{2}$

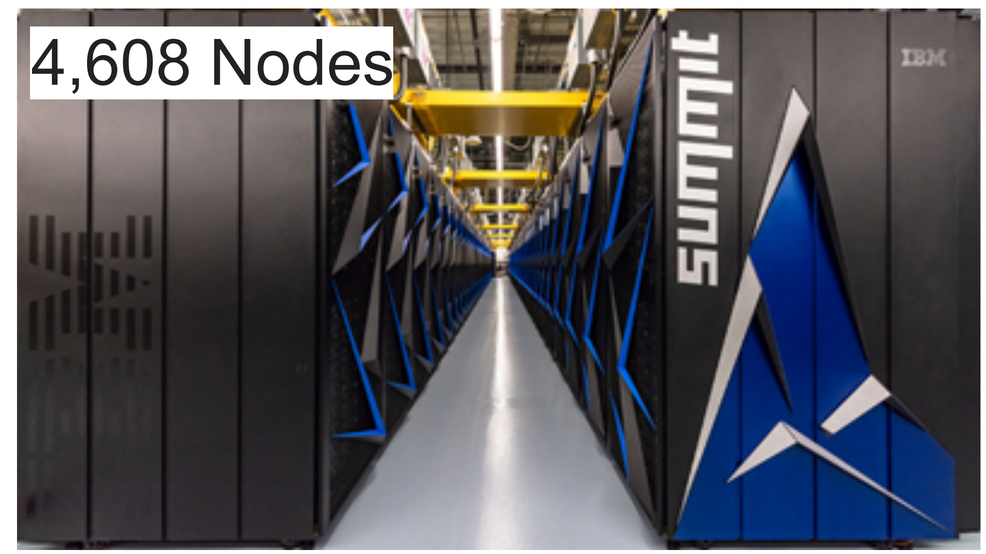
ASPL = **75**/45
= **1.67**...

Better !!

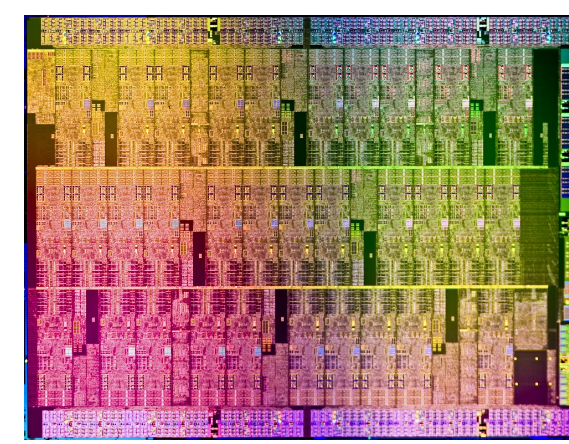
Background (2/3)

- ODP is useful for design of network topology with low latency since 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)

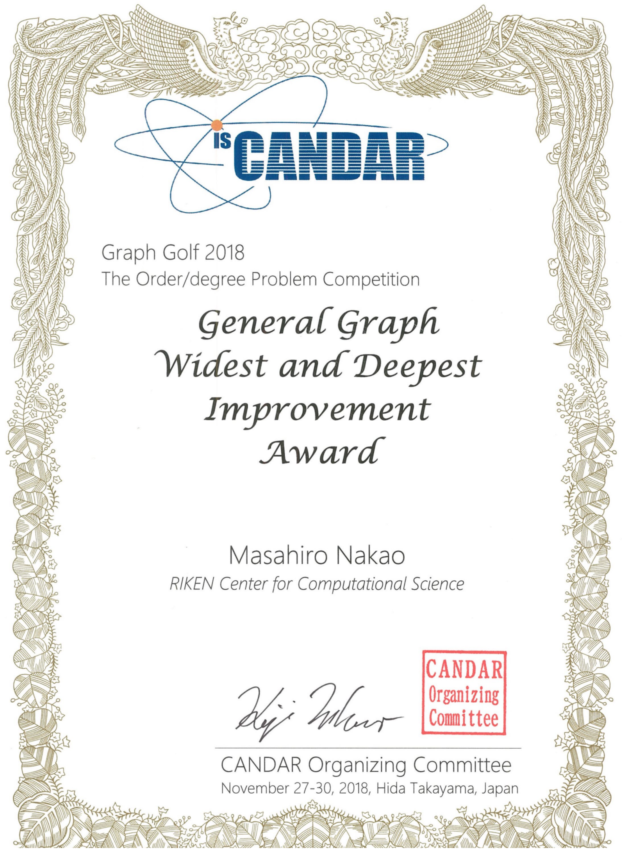
- **Graph Golf Competition for ODP**
  - http://research.nii.ac.jp/graphgolf
  - Held by National Institute of Informatics, Japan since 2015, every year
  - Rule
    - Find graphs with the smallest **diameter** and **ASPL** in problems
    - Problems are changed every year
    - Participants uploads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

14 problems in 2018

No	Problem (n, d)
1	72, 4
2	256, 5
3	256, 10
4	2300, 10
5	3019, 30
6	4855, 30
7	12000, 7
8	20000, 11
9	40000, 8
10	77000, 6
11	132000, 8
12	200000, 32
13	200000, 64
14	400000, 32

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 Competition in 2018



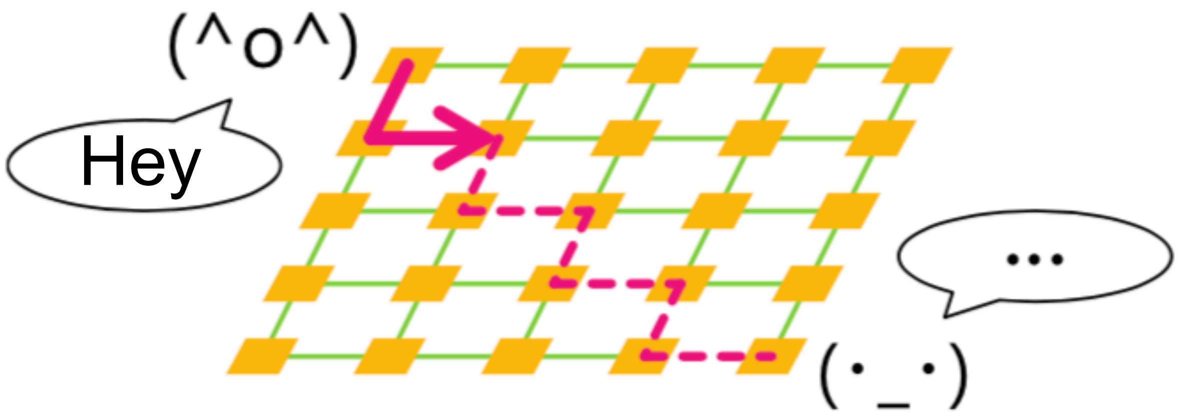
Award ceremony, CANDAR'18, Takayama, Gifu, Japan

Agenda

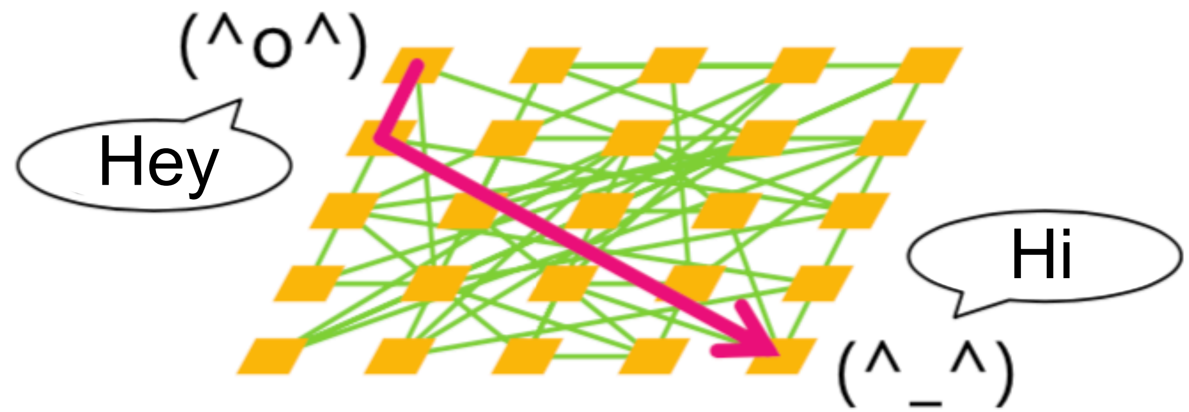
- Background
- Related works
- Explain our algorithm
- Evaluation of search and speed performances
- Summary

Related Works (1/2)

- Randomized topology [Koibuchi2012, Koibuchi2013, Singla2012]
  - When connecting nodes randomly, the distance shrinks due to the **small world effect**
    - Extensibility and failure resistance



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



Randomized Topology
(Shortcut allows a message to arrive soon)

https://www.nii.ac.jp/userimg/openhouse/2015/B11.pdf (Partial modification)

Related Works (2/2)

- Are randomized topologies best ? -> No

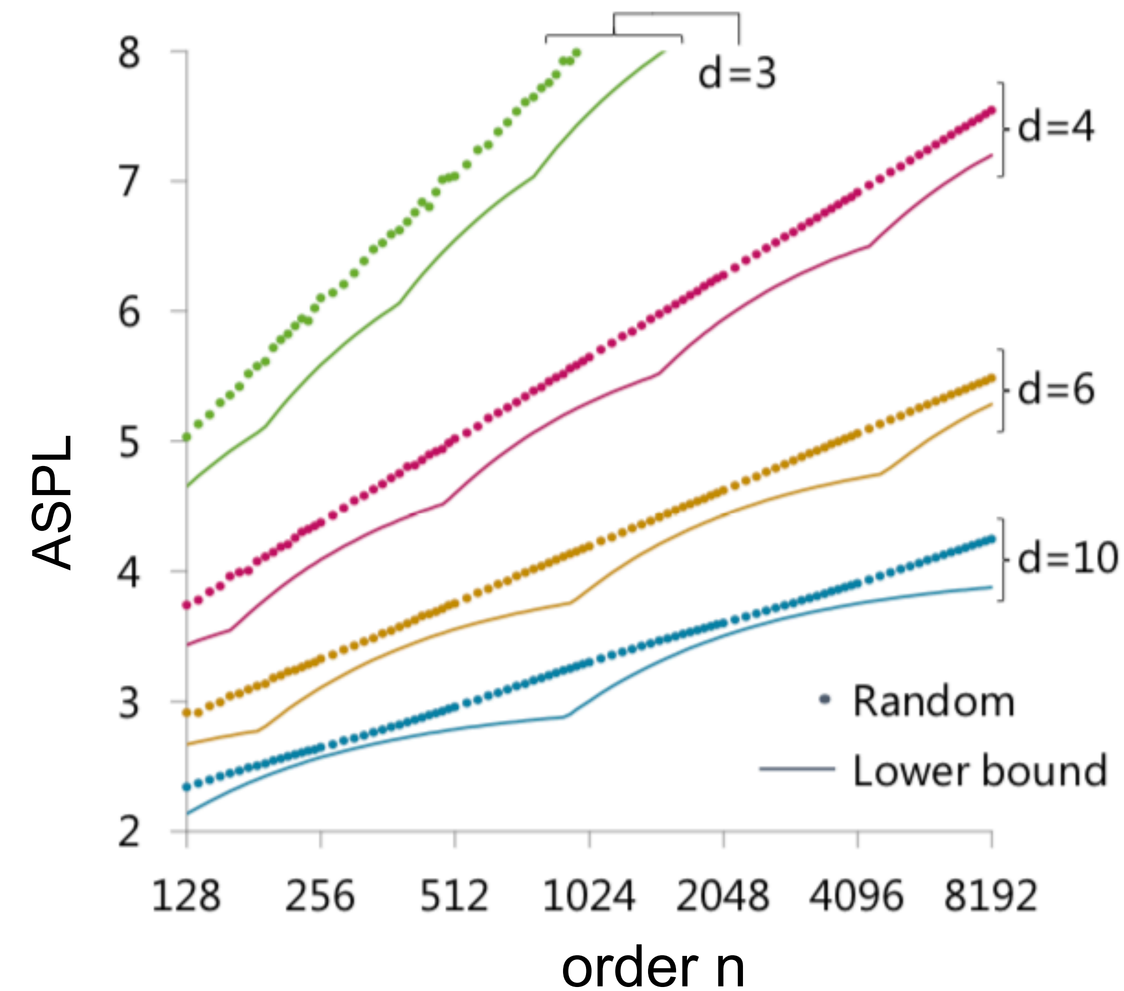
From the order n and degrees d,
the theoretical diameter lower bound K_{n,d}
and the ASPL lower bound L_{n,d} are calculated.

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

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

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

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



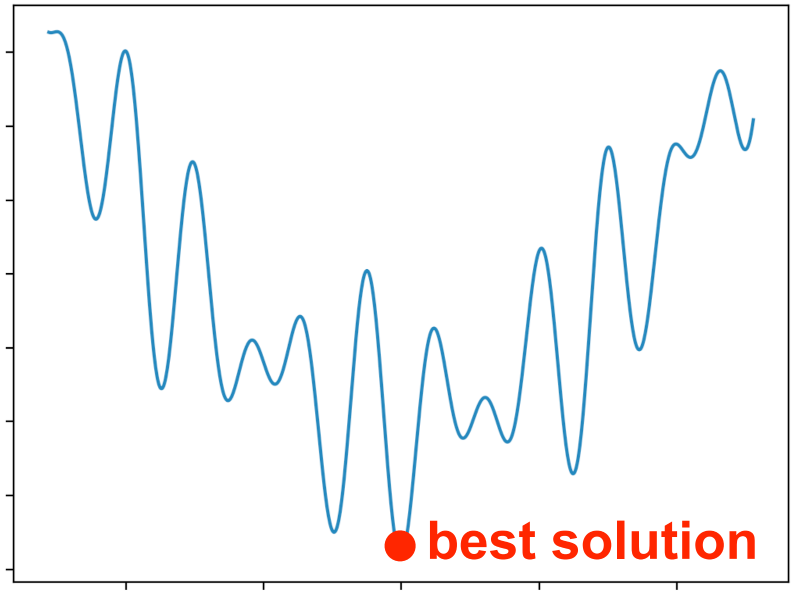
http://research.nii.ac.jp/graphgolf/2016/candar16/graphgolf2016-fujiwara.pdf

There is a gap between random topology and the theoretical lower bound

What are difficult points in ODP ?

(1) The number of graphs satisfying the given order n and degree d is enormous

- Number of graphs satisfying (n, d) = (12, 4) is about 480 billions
- It is difficult to find the **best solution** because a landscape of ODP's search space has many local optima



(2) The calculation time required for obtaining ASPL is enormous

- The calculation complexity with order n and degree d is **O(n^2*d)**
- 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 are calculated at the same time as ASPL)

Agenda

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

Origin of idea

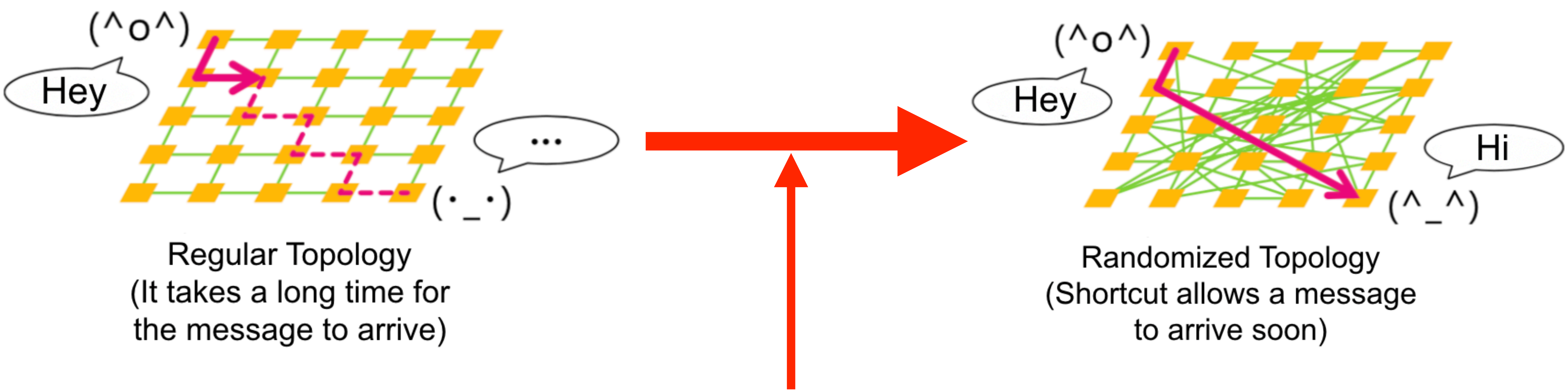
- What kind of topology does small diameter and ASPL have ?
  - Randomized topology is better than regular topology for latency
  - But, randomized topology is not best

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
  - SA is a one of random search algorithms, and the reason why SA is used that it doesn't easily fall into local optima
  - Reduce calculation time of ASPL (**Up to 10,000 times** with (n, d)=(400000, 32))

(2) Reducing the calculation time of ASPL further

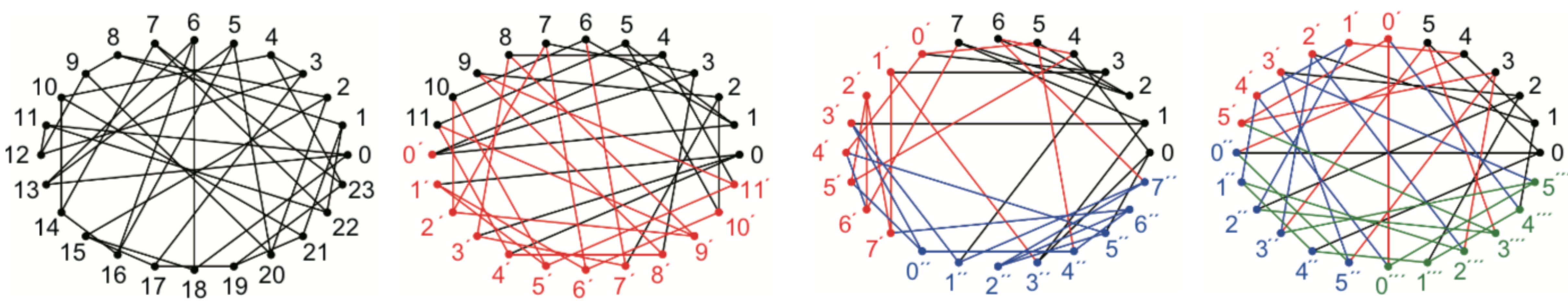
- Hybrid parallelization with MPI and OpenMP on our cluster system (**Up to 200 times** on 400 CPU cores with (n, d)=(400000, 32))
- Calculation time of ASPL decreased from 5.5 hours to 0.01 sec.

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

Our source code is available at
https://github.com/mnakao/GraphGolf

Graph symmetry

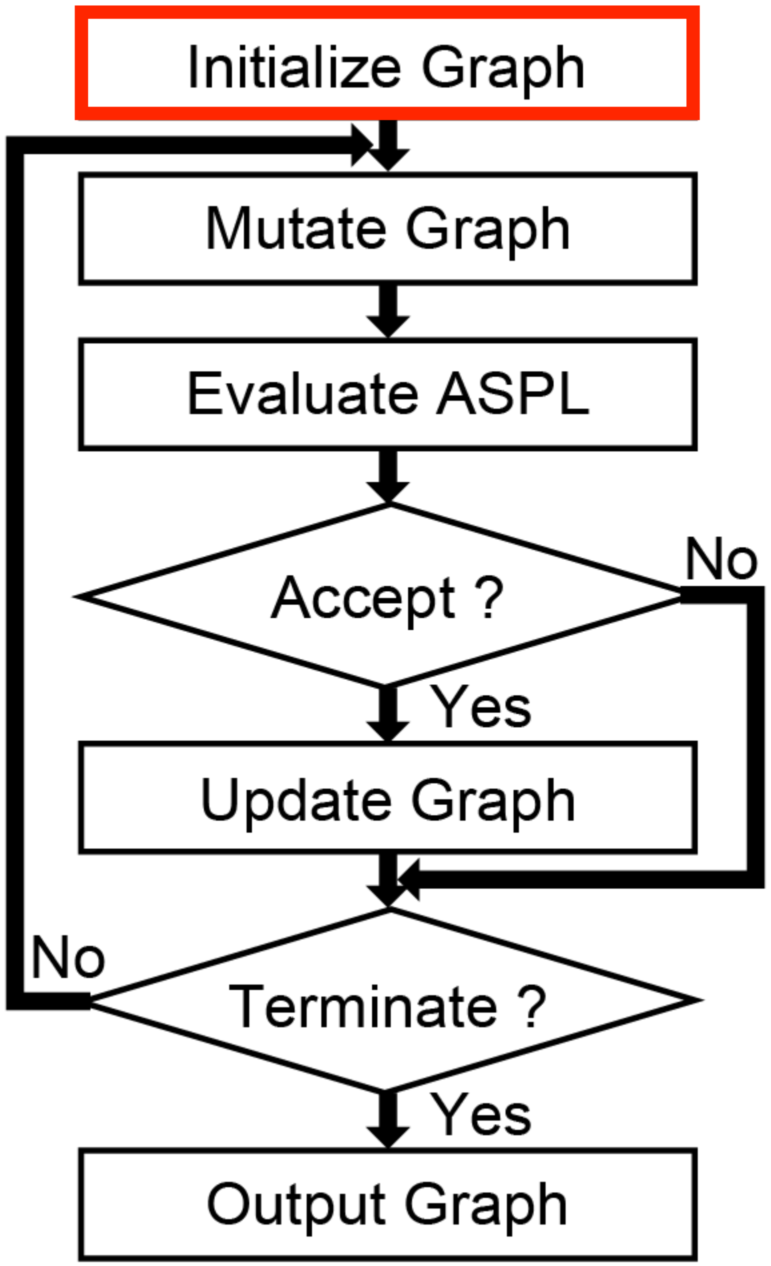
- Examples of the graph symmetry with (n, d) = (24, 3)



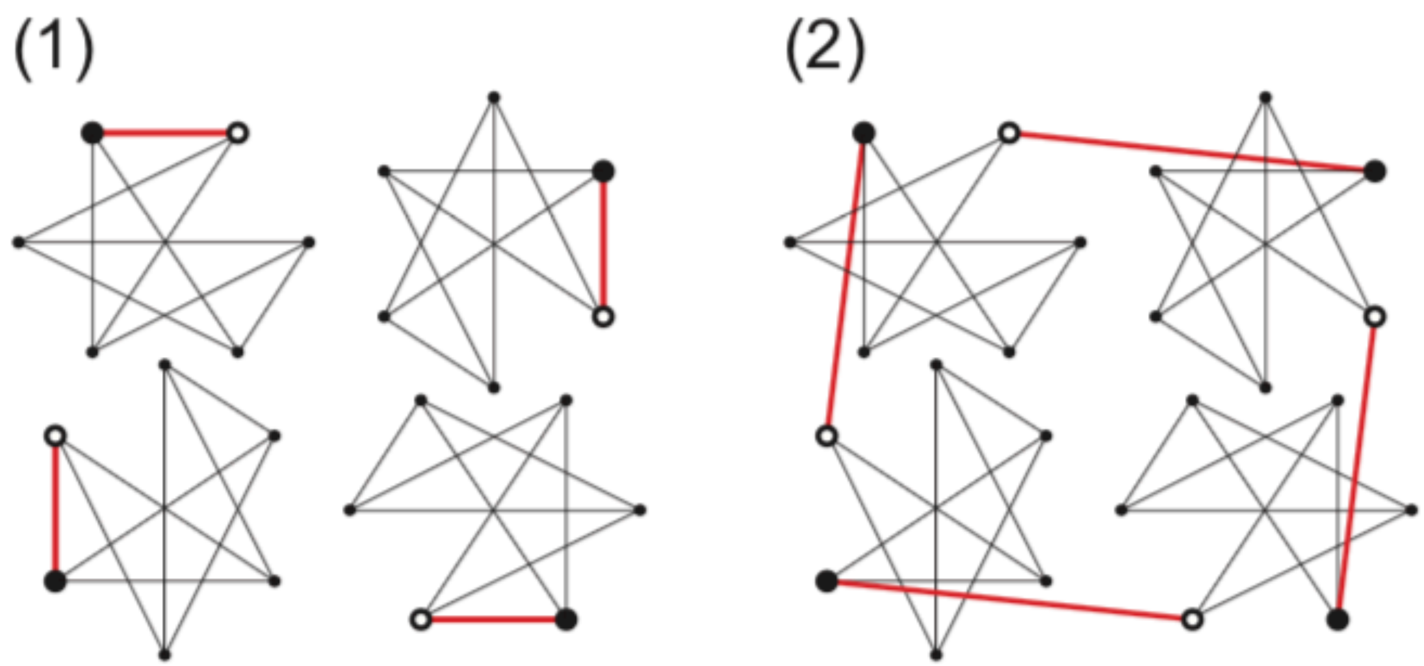
g=1 **g=2** **g=3** **g=4**

- 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



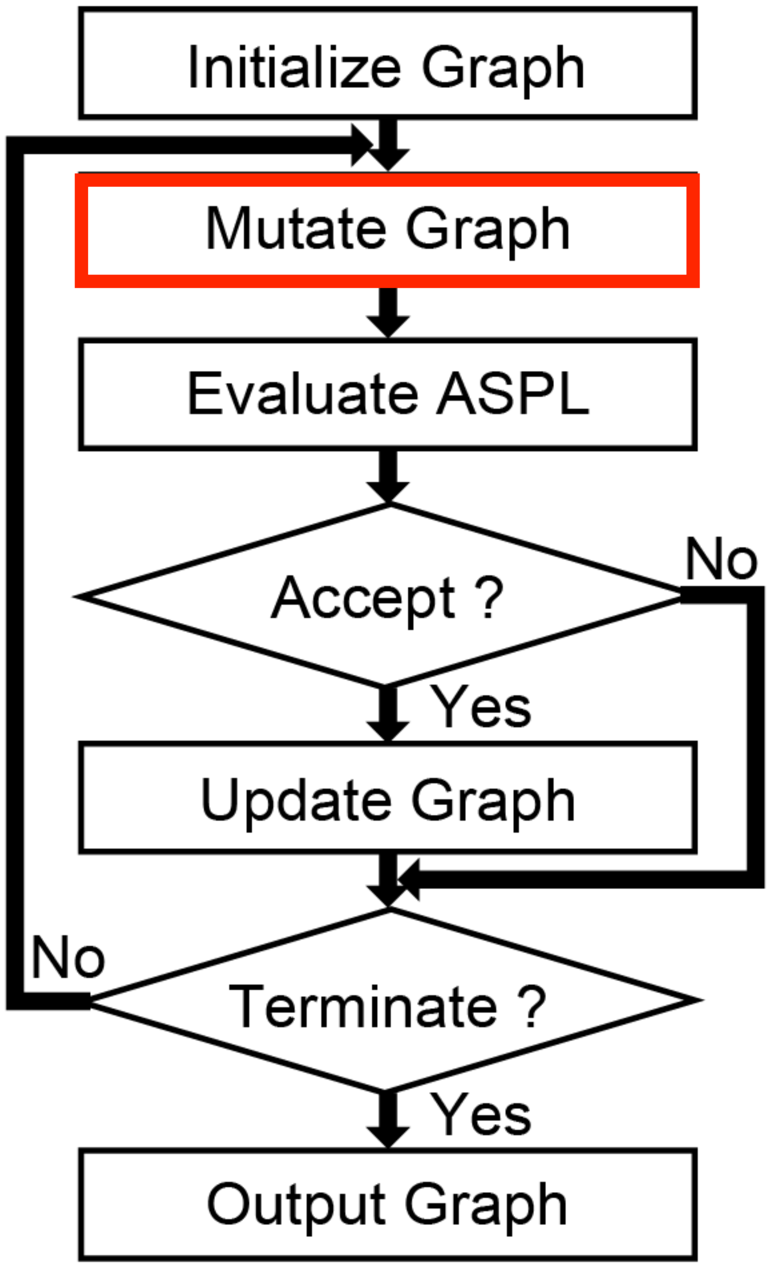
- Initialize a solution (n, d, g) = (24, 3, 4)



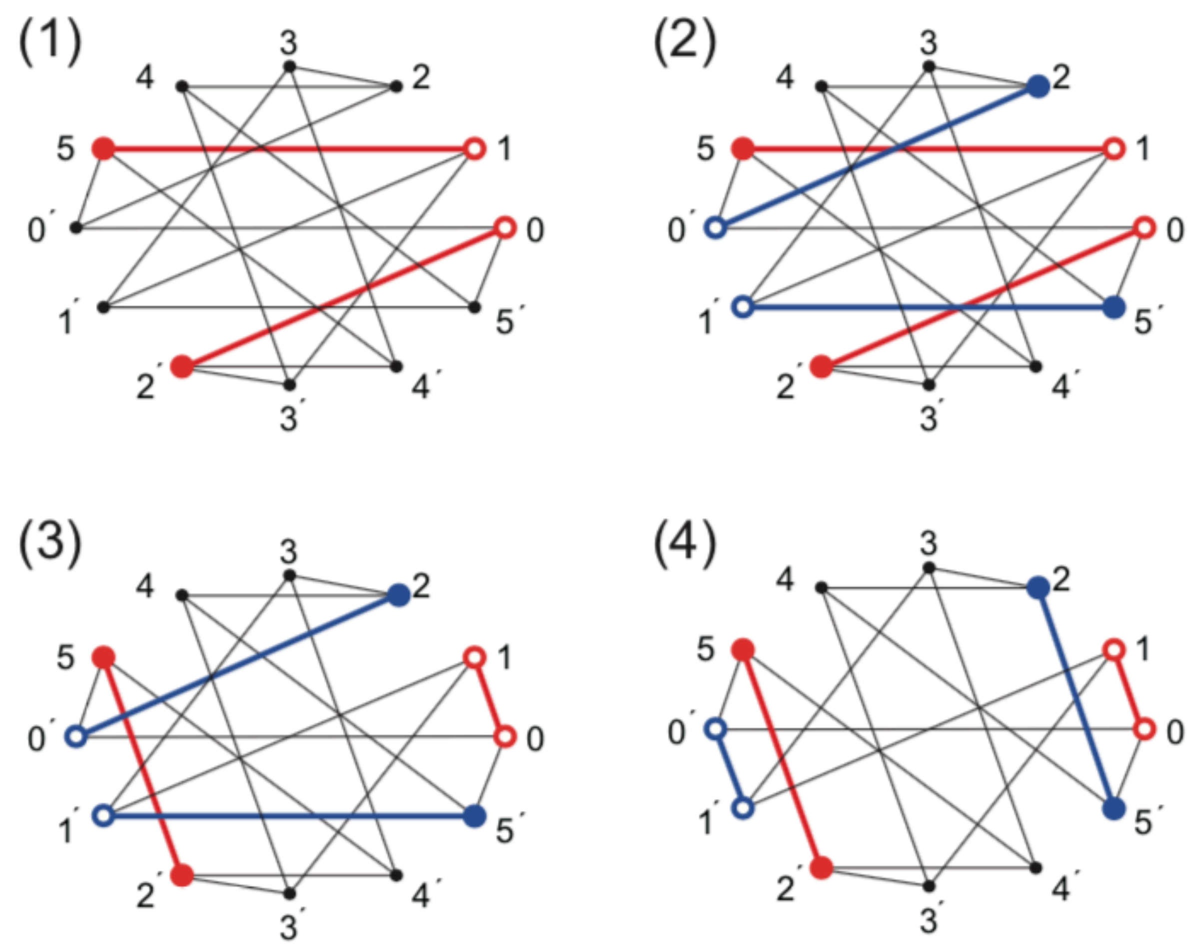
(1) Create a random graph with the number of nodes of the target graph divided by g, and duplicate g graphs (the graph with (n, d) = (6, 3) is created x 4). And select one edge from each graph symmetry.

(2) Connect both sides so that it becomes symmetrical

SA with Graph symmetry



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



(1) Randomly select two edges from all the edges

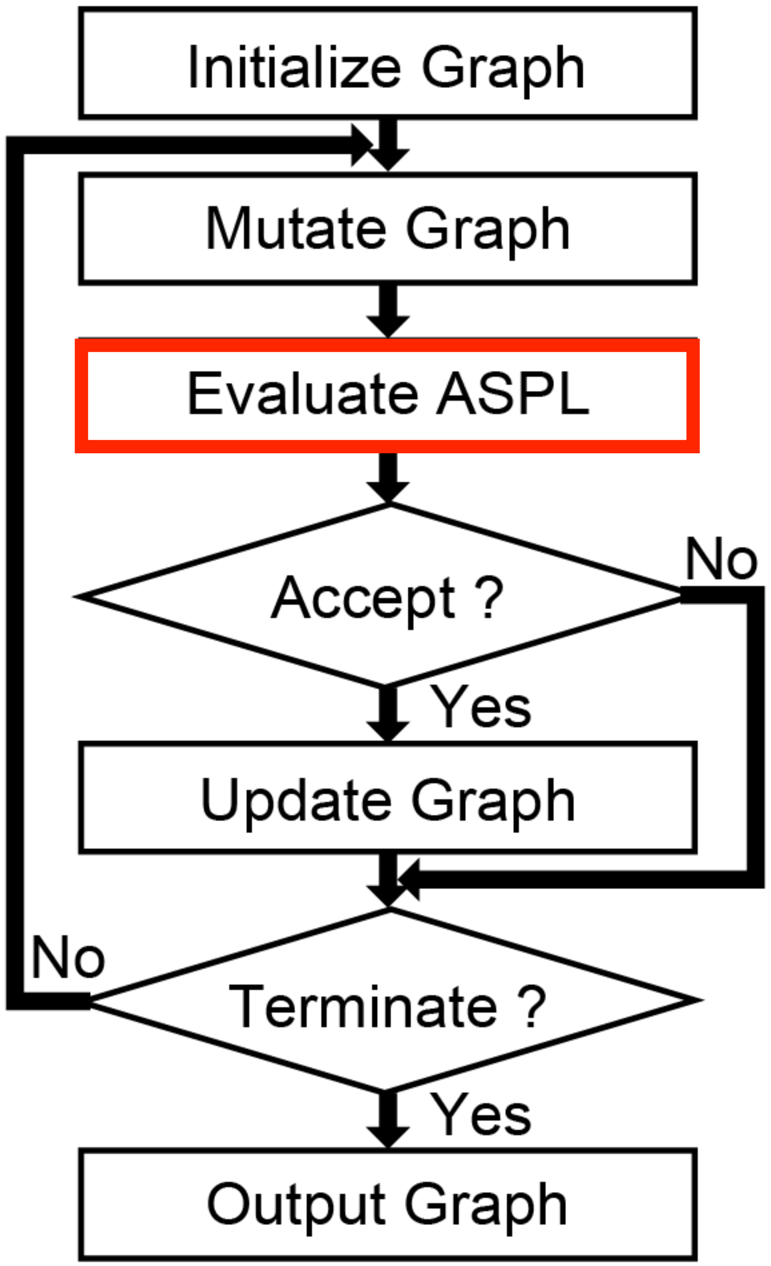
(2) Select edges symmetrically related to (1)

(3) Exchange edges of (1)

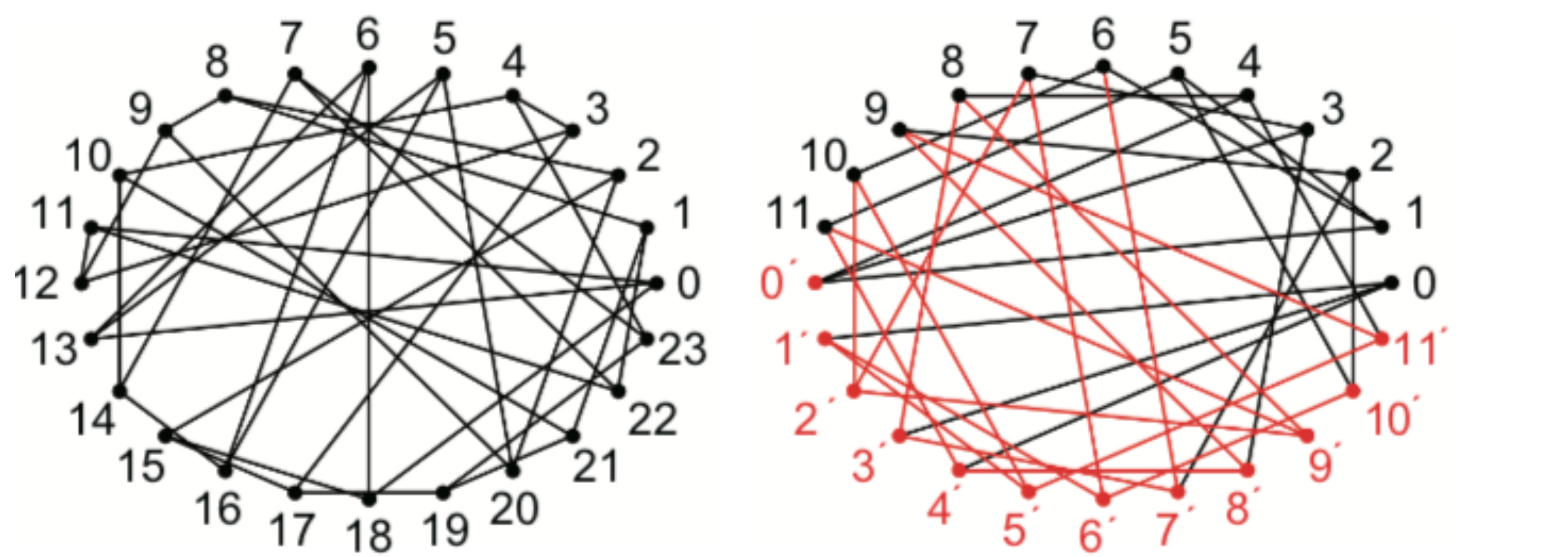
(4) Exchange edges of (2) in the same way as (3)

By performing the operation, a new solution is generated while maintaining symmetry

SA with Graph symmetry



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



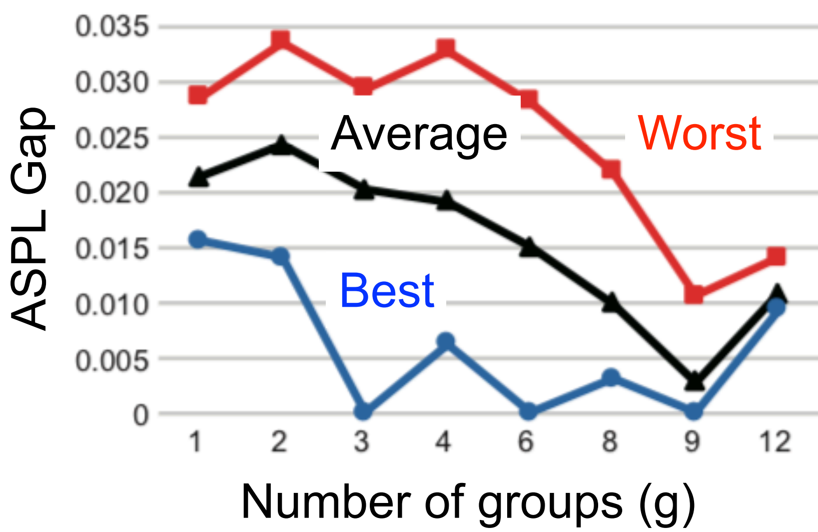
g=1 g=2

Agenda

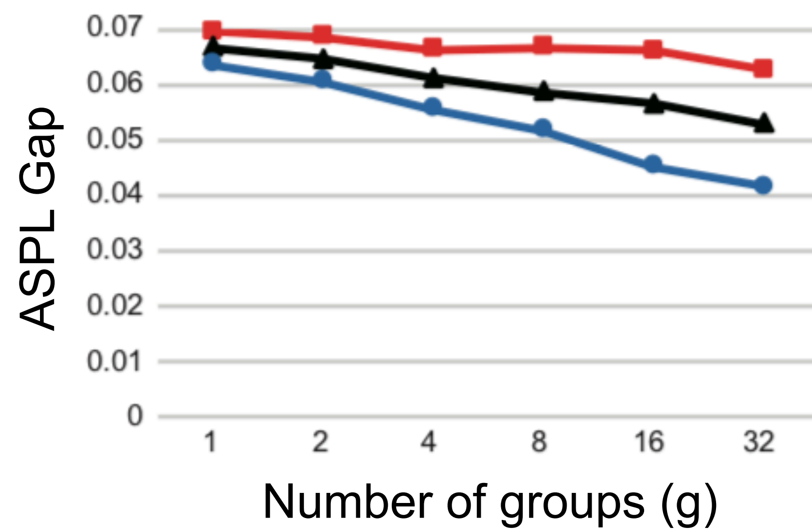
- Background
- Related works
- Explain our algorithm
- Evaluation of search and speed performances
- Summary

Search Performance

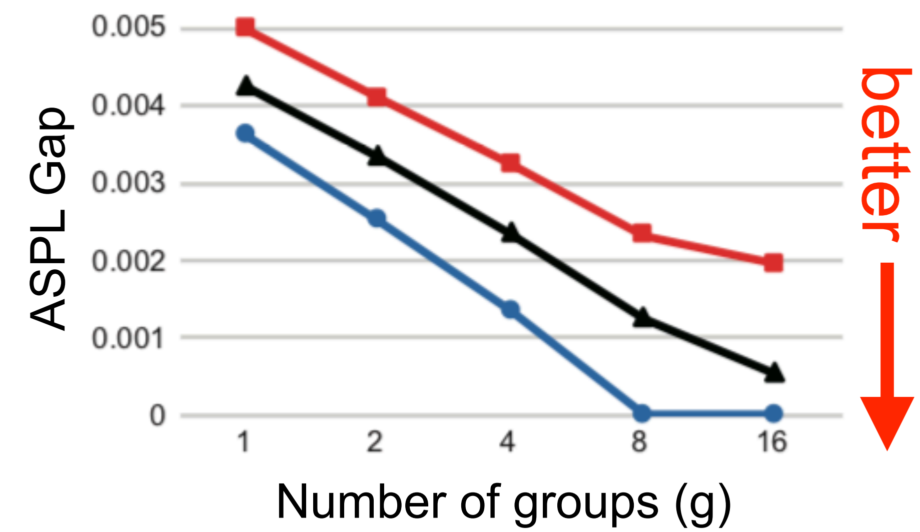
- Proposed method is executed 100 times with different g



(n, d) = (72, 4)



(n, d) = (256, 5)

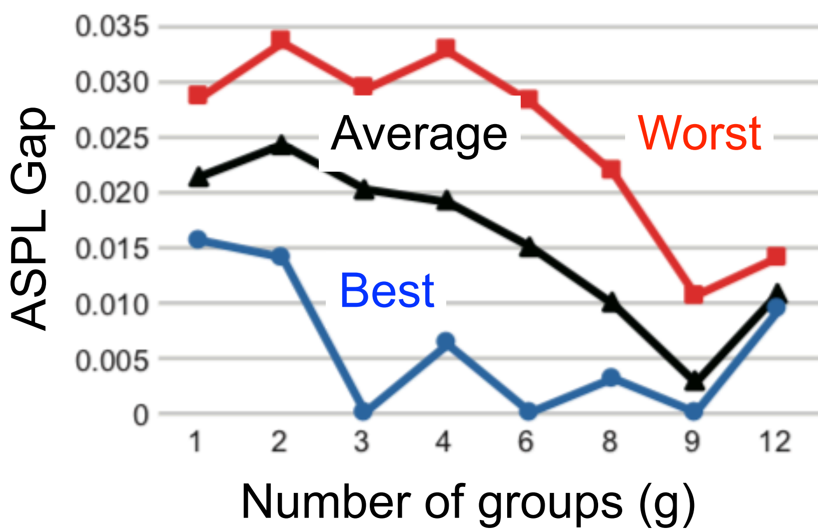


(n, d) = (256, 10)

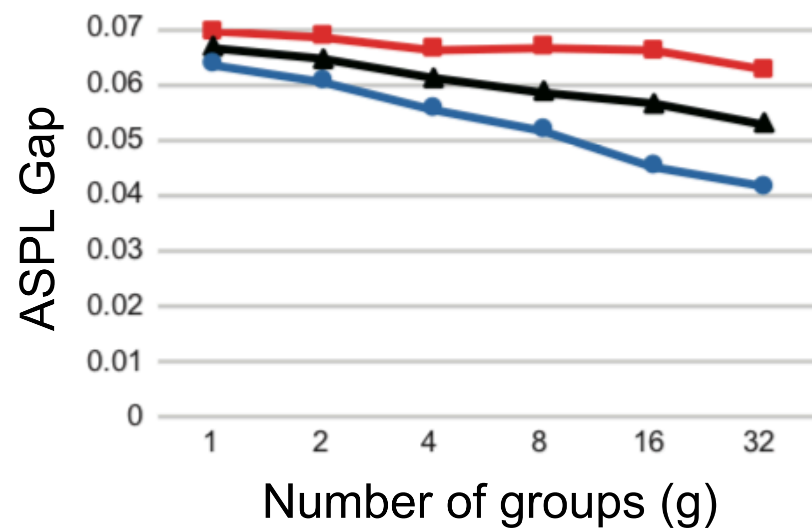
- In vertical axis, **ASPL Gap** is a difference from theoretical lower bound
- Search performance tends to increase as the g increases
- However, the problem (n, d) = (72, 4), the search performance is better for g=9 than for g=12, indicating that search performance may decrease if the value of g is too large
  - The g expresses the strength of regularity of a graph; regularity becomes stronger as g increases

Search Performance

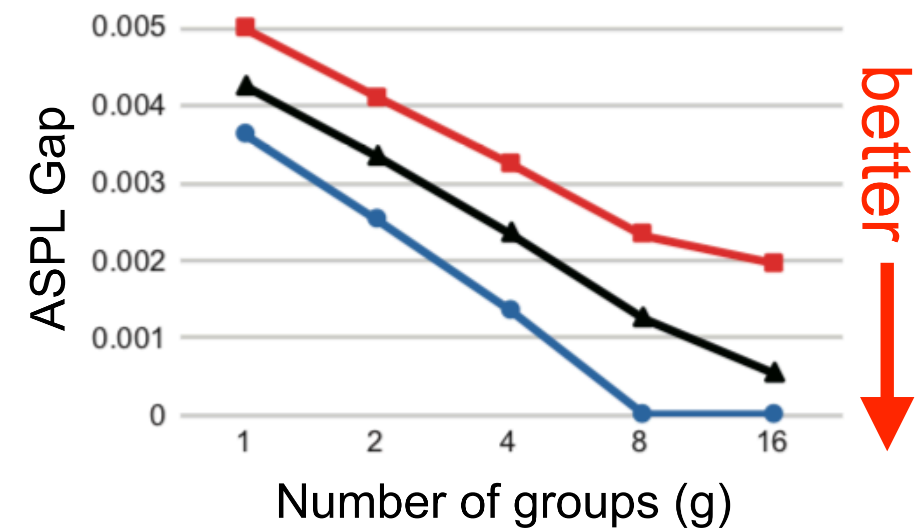
- Proposed method is executed 100 times with different g



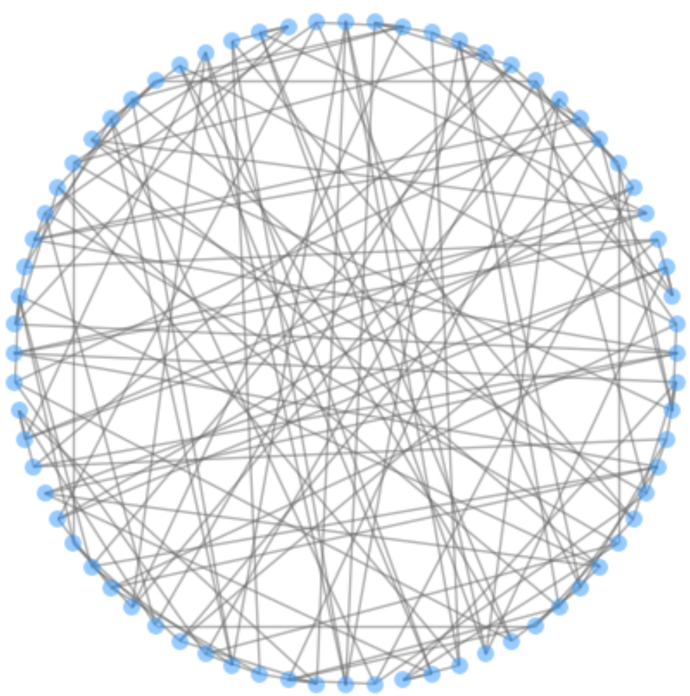
(n, d) = (72, 4)



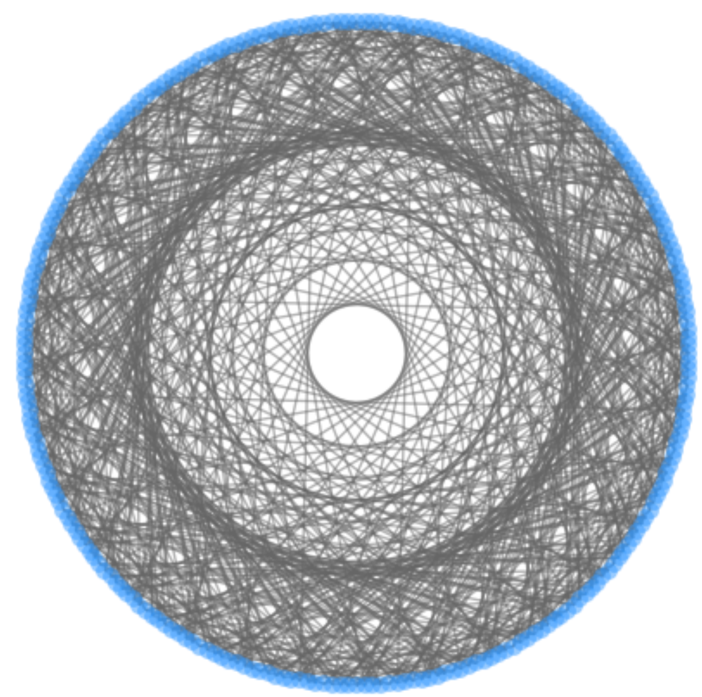
(n, d) = (256, 5)



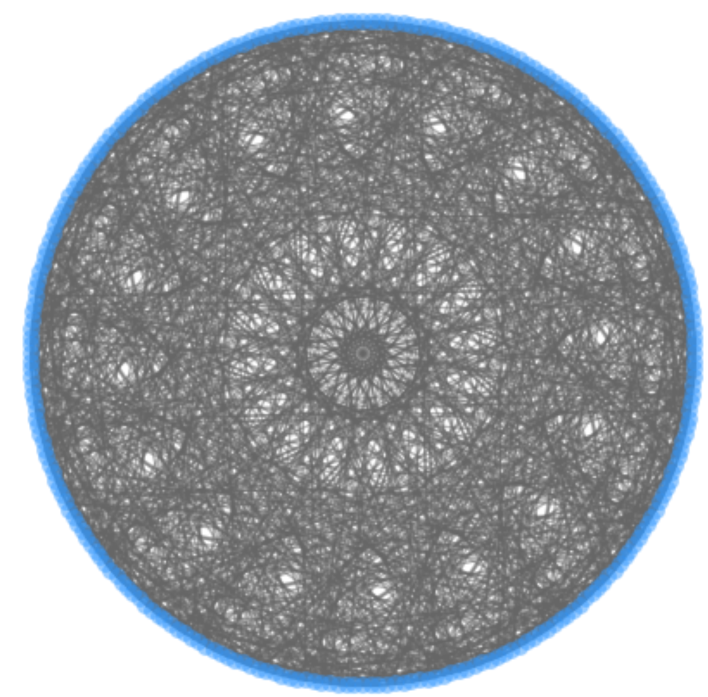
(n, d) = (256, 10)



Theoretical lower bound



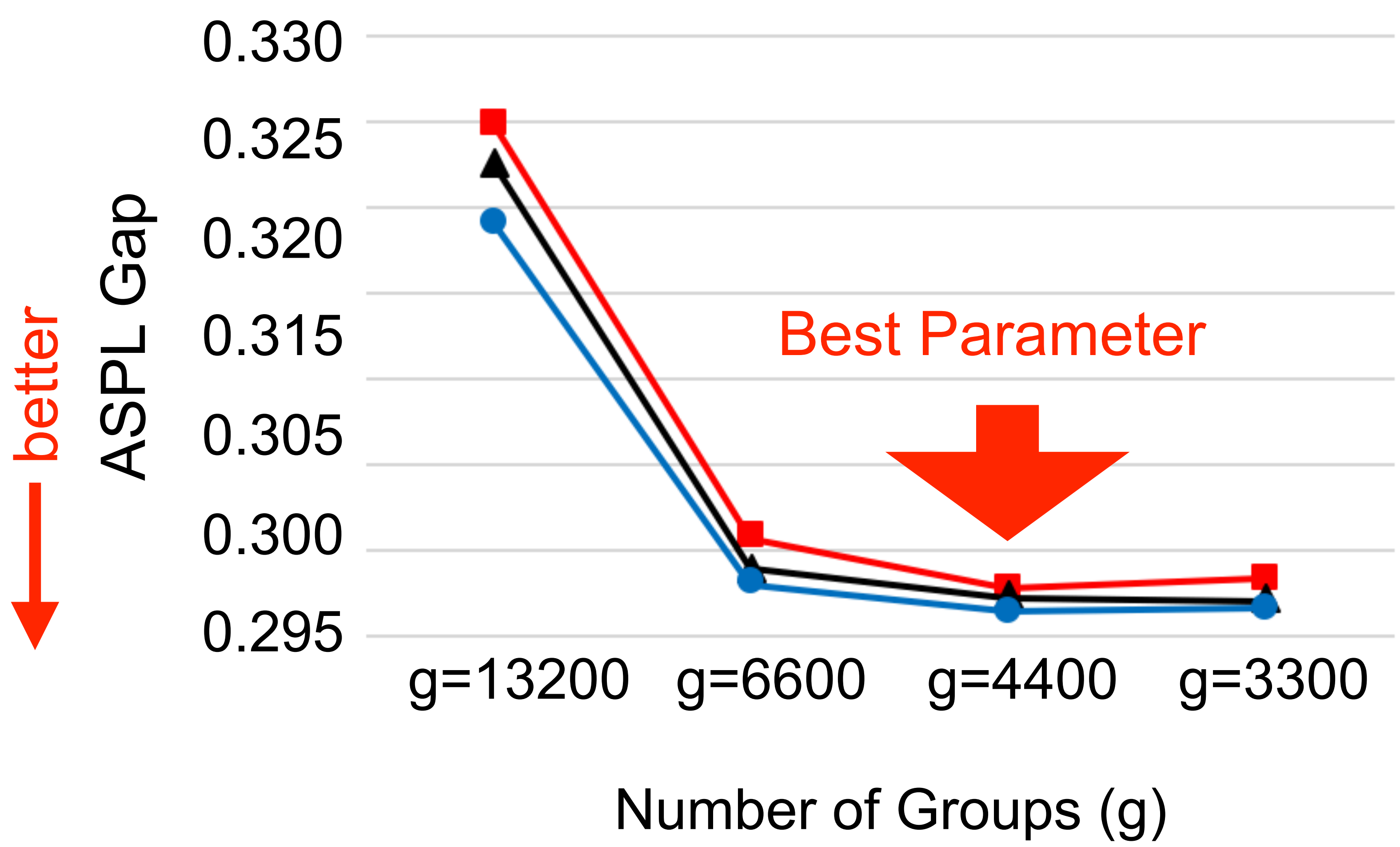
Best solution in Graph Golf



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)



The optimum g can be found by gradually reducing the value of g until ASPL Gap gets worse

Speed Performance

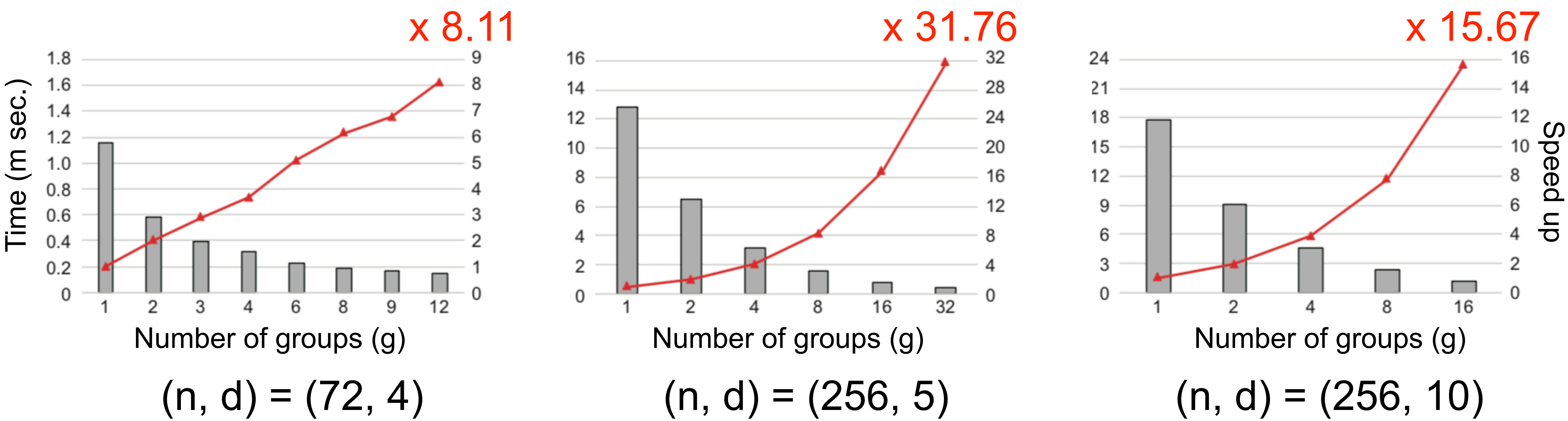
- COMA cluster system at University of Tsukuba

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

- Speed-up techniques
  - Graph symmetry
  - Hybrid parallelization with MPI and OpenMP

Performance results by Graph Symmetry

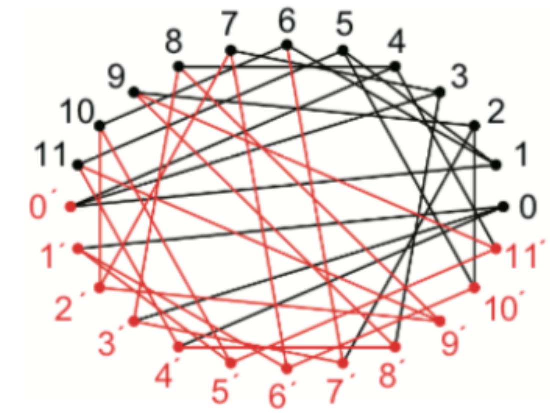
- Measure time to calculate ASPL 100 times



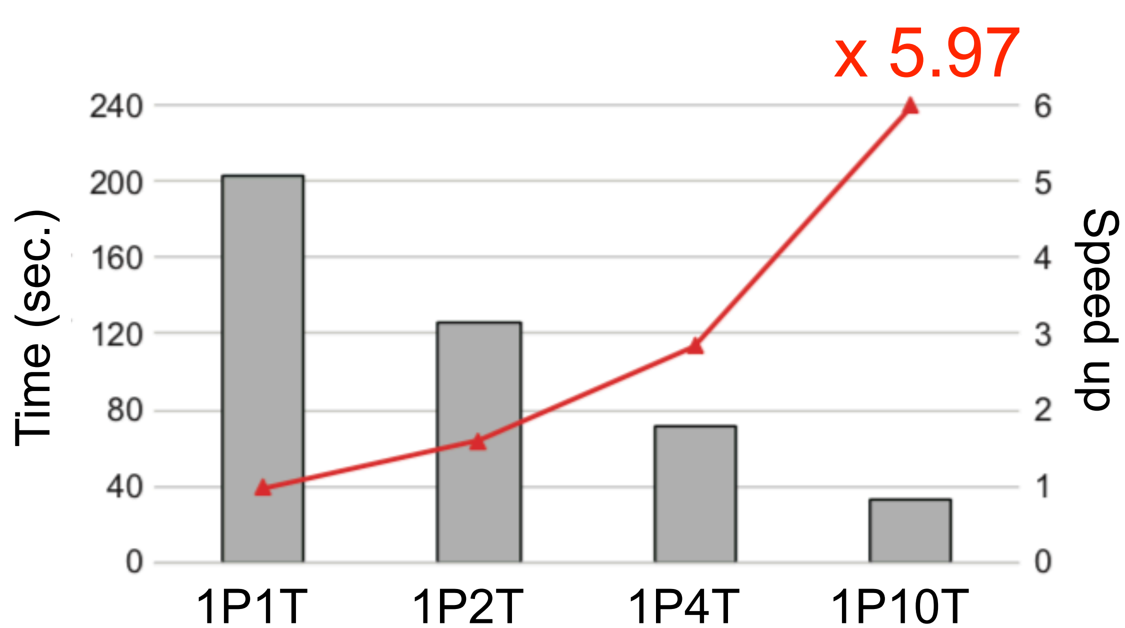
- The bars show the time on the left vertical axis, and the lines show the speed up ratio with g = 1 on the right vertical axis
- Speed ups of **8.11**, **31.76**, **15.67** times, respectively, were achieved for (n, d, g) = (72, 4, 12), (256, 5, 32), and (256, 10, 16)
  - The larger g, the smaller the time for ASPL

Performance results by Hybrid Parallelization

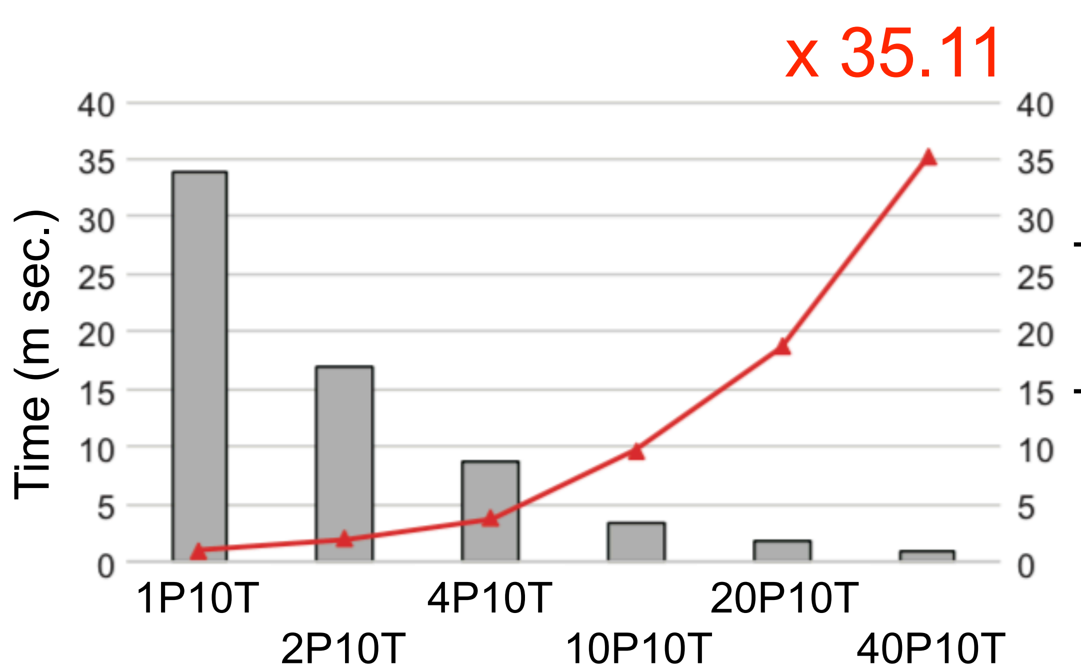
(n, d, g) = (24, 3, 2)



- Multiple BFS are executed using MPI at the same time, and each BFS is executed in parallel using OpenMP threads
- The maximum number of processes is (n/g)
  - In this case, each rank is in charge of one node
- In problem (n,d,g) = (400000,32,10000), the performance with 40 procs. and 10 threads is about **200** (5.97 x 35.11) times faster than that with 1 proc. and 1 thread



Thread parallelization with OpenMP



Hybrid parallelization with MPI and OpenMP

Results of Graph Golf 2018

🏆: Our algorithm is best

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

Proposed method won
8 problems in 14 problems

Agenda

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

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 using hybrid parallelization with MPI and OpenMP, the calculation time for the ASPL was further reduced
  - A performance improvement of about **200** 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 **2,000,000** times compared to that obtained with g=1
    - The calculation time of ASPL decreased from 5.5 hours to **0.01 seconds**