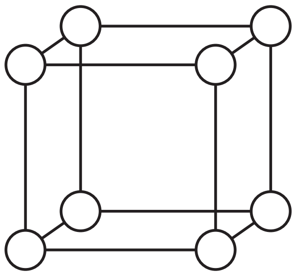
グラフゴルフのはじめ方
手書きグラフから
経験的手法によるグラフ生成まで

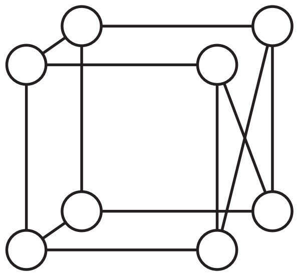
北須賀 輝明, 飯田 全広

熊本大学

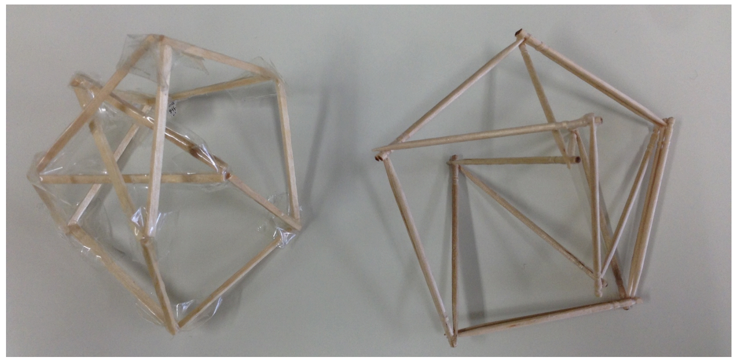
大学院先端科学研究部



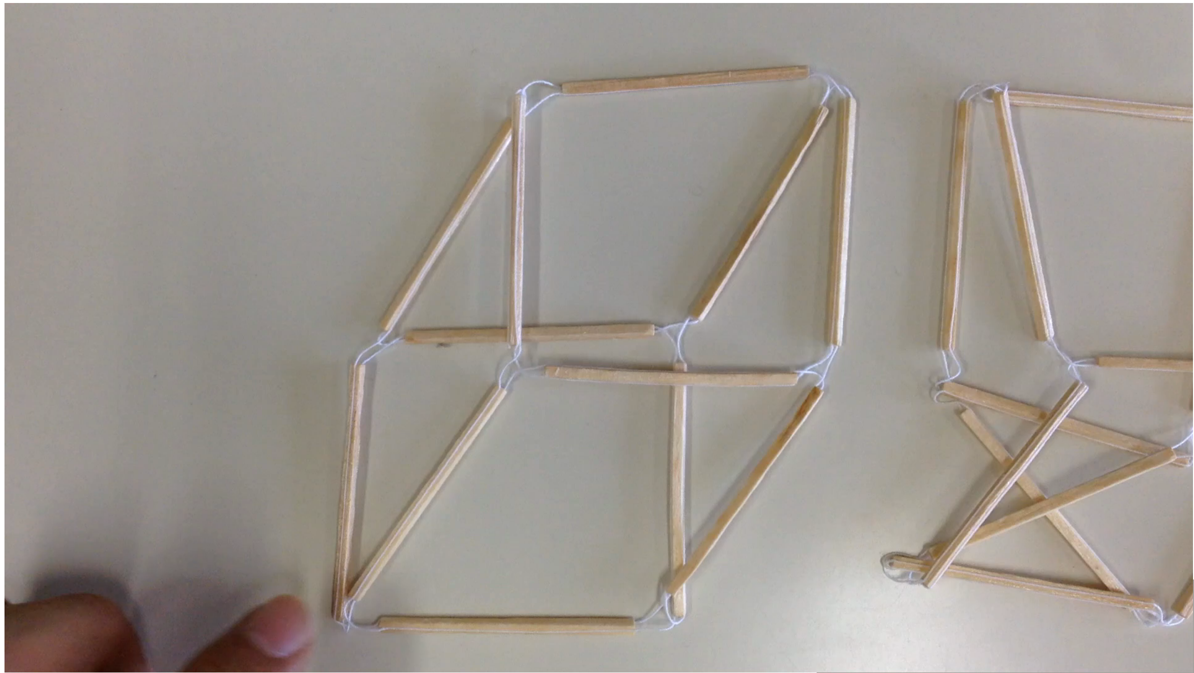
Cube; k=3, l=1.71 (Gap 9.1%)



Möbius loop; k=2, l=1.57 (Gap 0.0%)

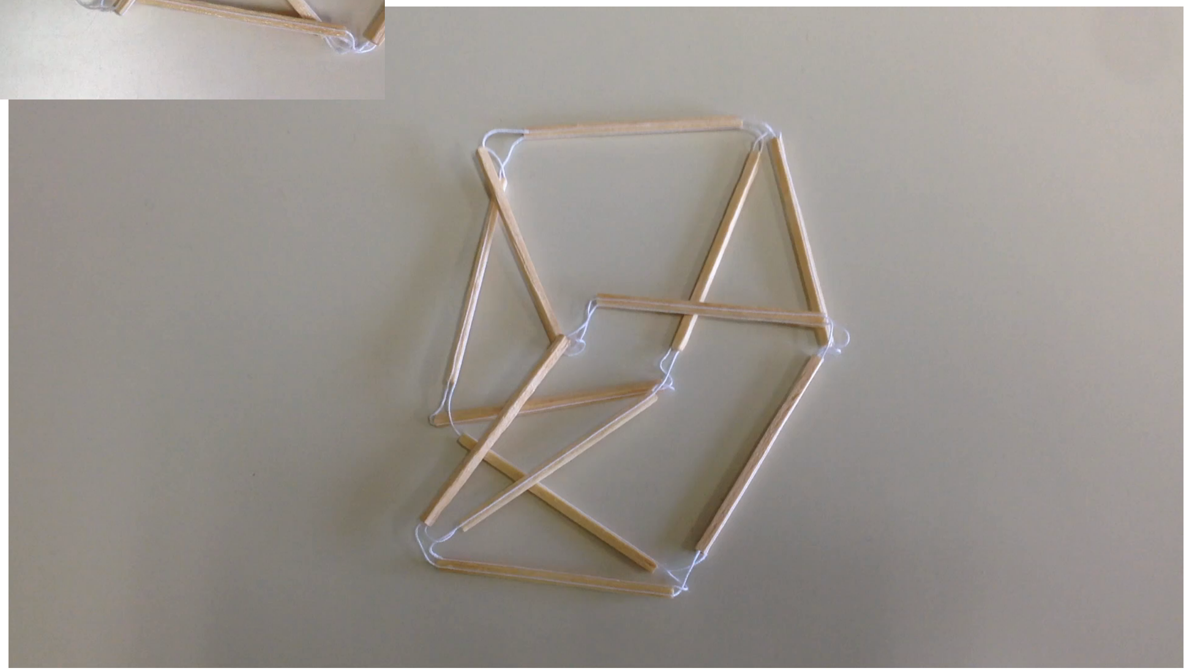


自然と直径が求まる手作りグラフ
（$n = 8$, $d = 3$）



Cube

($k = 3$, $l = 1.71$)



Möbius loop

($k = 2$, $l = 1.57$, 最適解)

概要

- Graph Golf 2015とその後の成果
- 手書きグラフの観察
- 経験的手法による直径3のグラフ生成
  - ✓グラフ G_0 の生成と貪欲な枝の追加
- 2-Opt局所探索
  - ✓枝の重要度
- まとめ

Home Problem Solutions Submit Event Q&A About

Graph Golf

The Order/degree Problem Competition

> *Find a graph that has smallest diameter & average shortest path length given an order and a degree.*

Best solutions

Update 2015-10-06

Degree d	Order n 16	64	256	4096	10000
3	3 / 2.200 0.000%	5 / 3.770 0.211%	8 / 5.636 0.861%	13 / 9.787 2.928%	15 / 11.122 3.225%
4	3 / 1.750 0.962%[2]	4 / 2.869 0.417%	6 / 4.134 1.065%	9 / 6.756 4.423%	10 / 7.601 3.480%
16	N/A	2 / 1.746 0.000%	3 / 2.093 8.026%[2]	4 / 3.254 8.768%	5 / 3.626 1.072%
23	N/A	2 / 1.635 0.000%[1]	2 / 1.910 0.000%	4 / 2.887 0.752%	4 / 3.201 8.697%
60	N/A	2 / 1.048 0.000%[1]	2 / 1.765 0.000%[1]	3 / 2.295 8.976%	3 / 2.650 0.624%
64	N/A	N/A	2 / 1.749 0.000%[1]	3 / 2.242 12.994%[2]	3 / 2.610 1.012%

Legend: Diameter / Average shortest path length (ASPL)
Gap from the lower bound of ASPL (%)

Notes: 1. A random graph is optimal. Submissions with this size will not be awarded.
2. There are no graphs with this size that satisfy the lower bound of diameter and ASPL [Erdös 1980].

Show complete list

solutions!

Motivation

扱う問題と扱わない問題

扱う問題

- 任意の2ノード間の（最短）経路長の最大長が短いネットワークを構成する
- 任意の2ノード間の（最短）経路長の平均長が短いネットワークを構成する

扱わない問題

- 経路（ルーティング）: 任意の2ノード間の最短経路を求める方法. （今後重要になります）
- 動的なトポロジ変更

Graph Golf 2015とその後の成果

平均パス長 /, 直径3

成果（現時点で最良）
- n = 4096, d = 60
- n = 4096, d = 64
- n = 256, d = 16 (コンペ後)
- n = 10000, d = 60 (コンペ後)

次数 d	ノード数 n 16	64	256	4096	10000
3	2.200 (July, *2) (= best known)				
4	1.750 (July, *2) (= best known)				
16	(N/A)		2.09069 (After c.., *3) (< 2.09262 best known)	(3.25426873 best known)	(3.62562128 best known)
23					
60				2.295216 (After c., *1) 2.295275 (Aug.)	2.648977 (After c., *3) 2.648980 (After c.) (2.650399 best known)
64				2.242170 (Oct.,, *1) 2.242228 (Sep.,)	2.611310 (After c.) (> 2.610117 best known)

Diameter k (lower bound; best known)

degree d	order n 16	64	256	4096	10000
3	3; 3	5; 5	7; 8	11; 13	12;15
4	2; 3	4; 4	5; 6	7; 9	8; 10
16	N/A	2; 2	2; 3	4; 4	4; 5
23	N/A	2; 2	2; 2	3; 4	3; 4
60	N/A	2; 2	2; 2	3; 3	3; 3
64	N/A	N/A	2; 2	2; 3	3; 3

Note:
- *1: best solution in comp.
- *2: best but late submission
- *3: best after competition

Graph Golf 2015とその後の成果
平均パス長 /, 直径3

次数 d	ノード数 n 16	64	256	4096	10000
3	2.200 (July, *2) (= best known)				
4	1.750 (July, *2) (= best known)				
16	(N/A)		2.09069 (After c.., *3) (< 2.09262 best known)	(3.25426873 best known)	(3.62562128 best known)
23					
60				2.295216 (After c., *1) 2.295275 (Aug.)	2.648977 (After c., *3) 2.648980 (After c.) (2.650399 best known)
64				2.242170 (Oct.,, *1) 2.242228 (Sep.,)	2.611310 (After c.) (> 2.610117 best known)

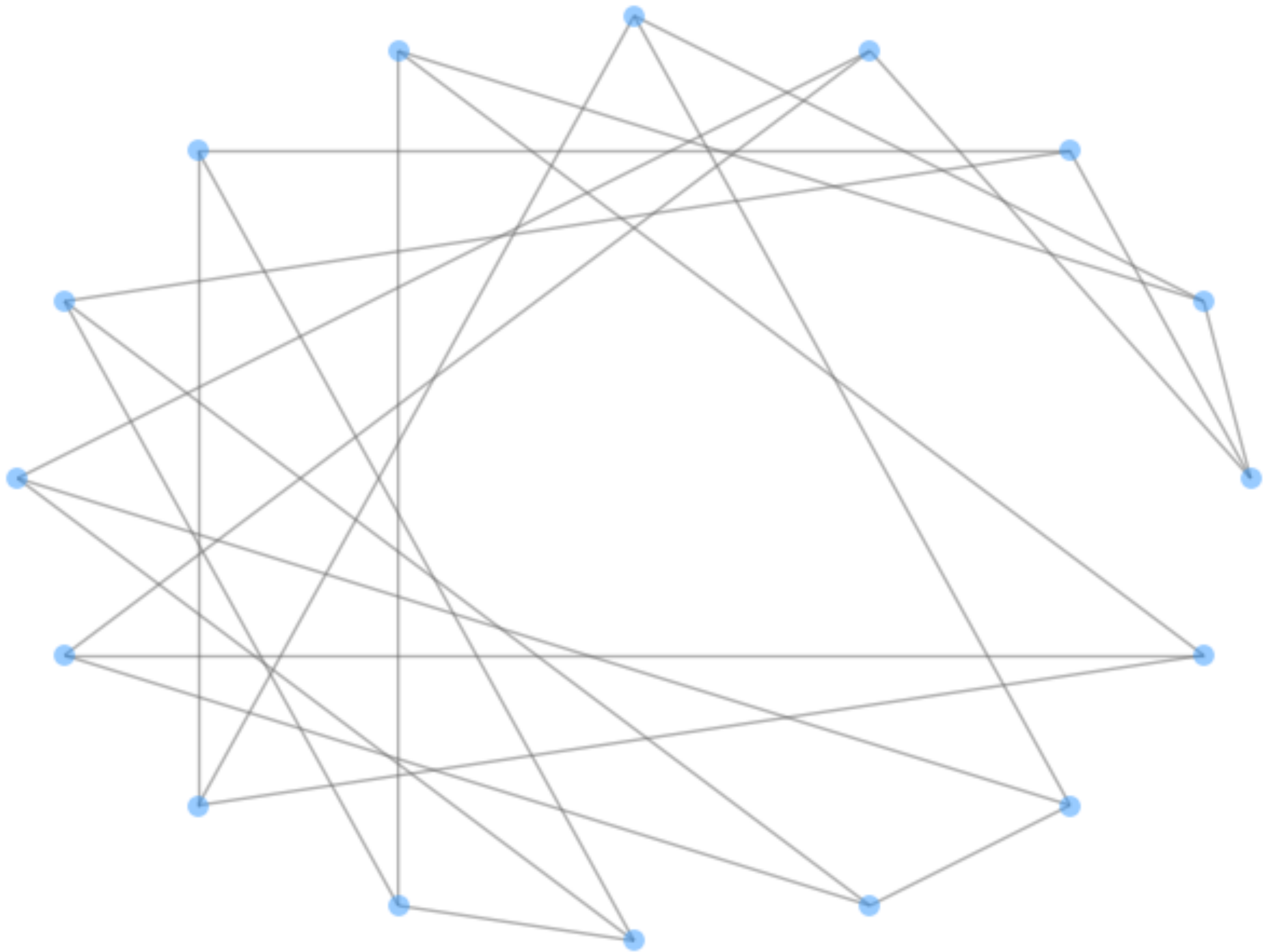
Diameter k (lower bound; best known)

degree d	order n 16	64	256	4096	10000
3	3; 3	5; 5	7; 8	11; 13	12;15
4	2; 3	4; 4	5; 6	7; 9	8; 10
16	N/A	2; 2	2; 3	4; 4	4; 5
23	N/A	2; 2	2; 2	3; 4	3; 4
60	N/A	2; 2	2; 2	3; 3	3; 3
64	N/A	N/A	2; 2	2; 3	3; 3

Note:
- *1: best solution in comp.
- *2: best but late submission
- *3: best after competition

グラフ例：$n = 16$, $d = 3$; $k = 3$, $l = 2.200$

(作者：Nobushimi & Ryo Ashida & Ryuhei Mori)



図：Graph Golf 2015のページより

グラフ例 : n = 16, d = 3; k = 3, l = 2.200
(作者 : H. Inoue)

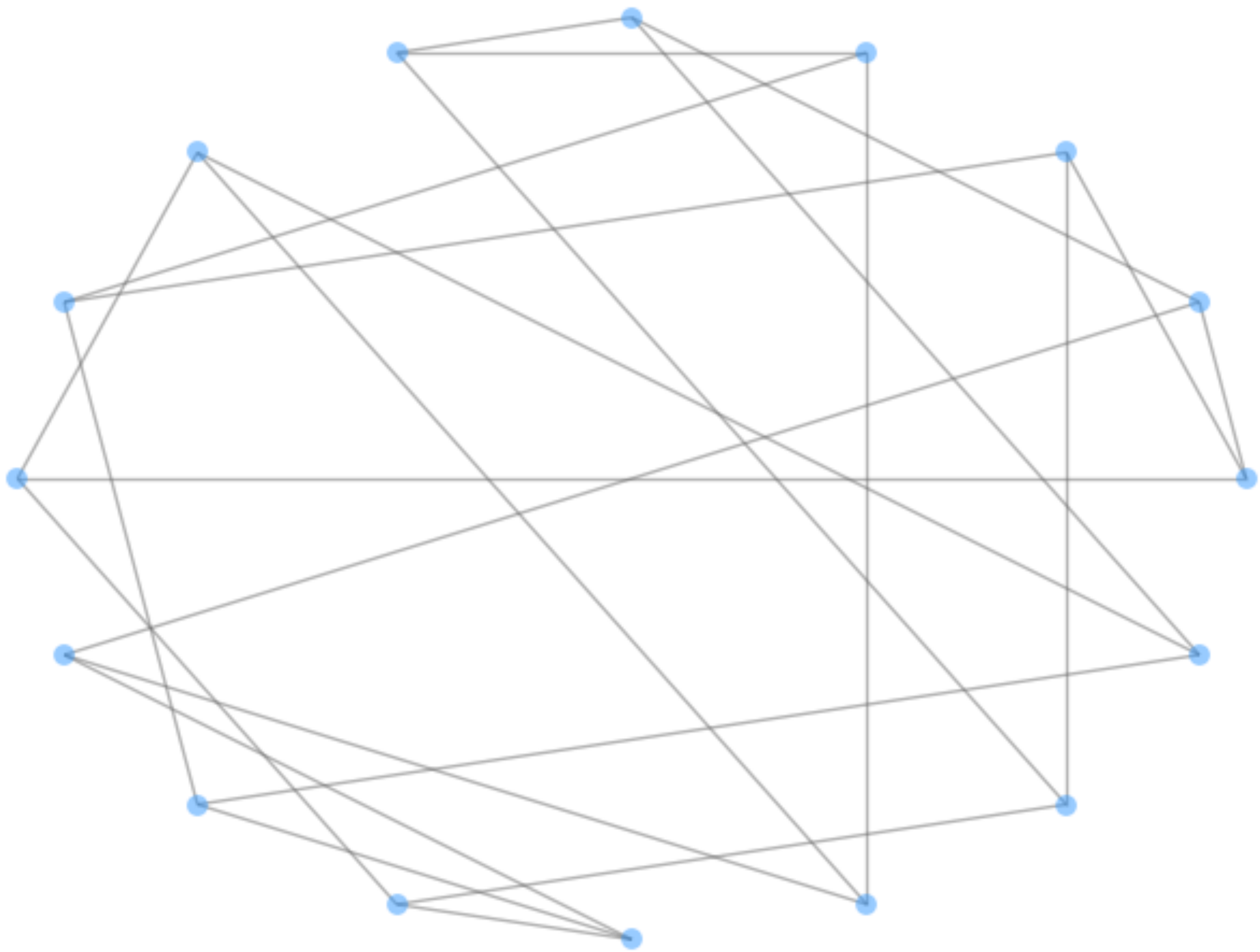


図 : Graph Golf 2015のページより

グラフ例 : $n = 16,\ d = 3;\ k = 3,\ l = 2.200$

(作者 : flowlight)

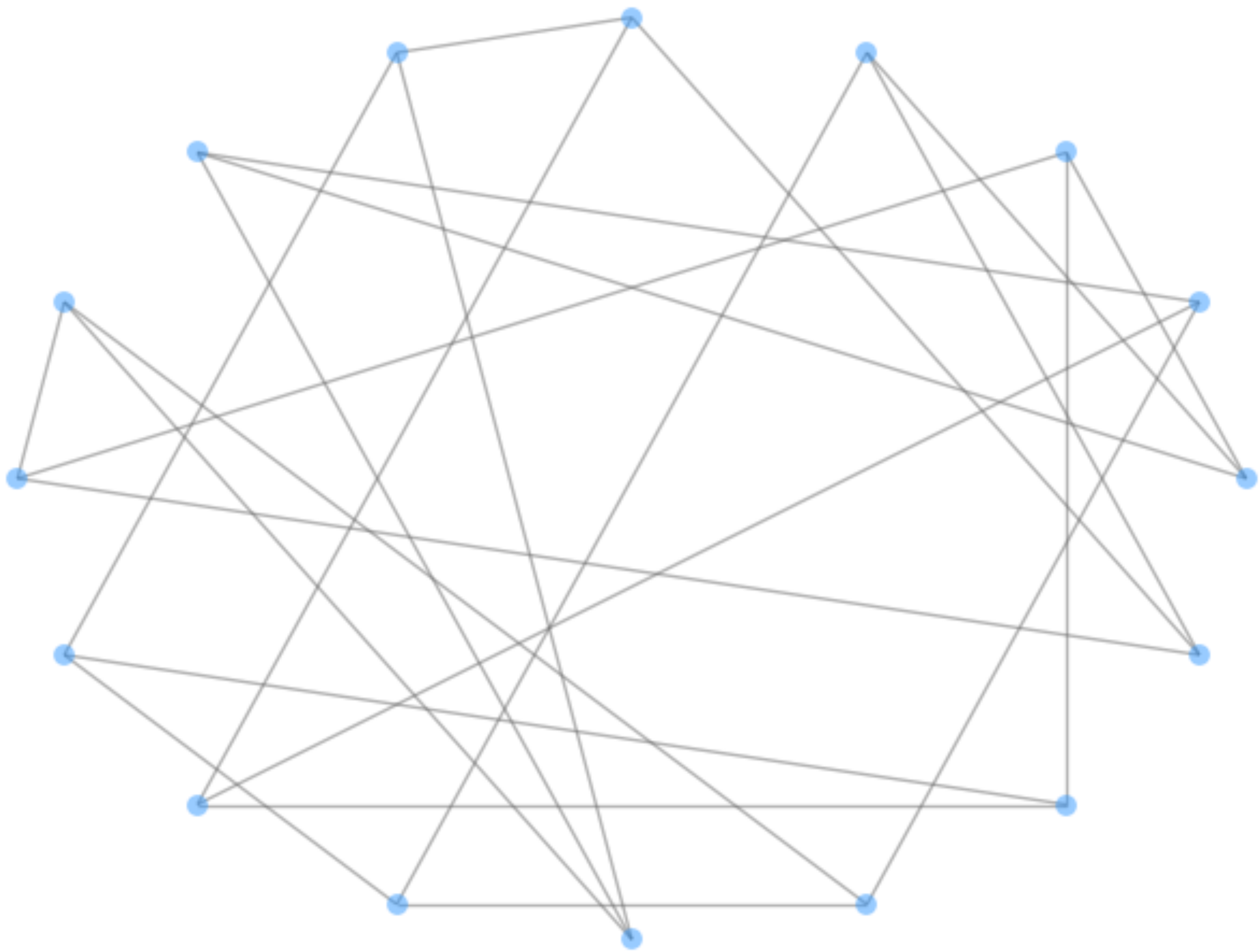


図 : Graph Golf 2015のページより

グラフ例 : n = 16, d = 3; k = 3, l = 2.200
(作者 : yawara & amami)

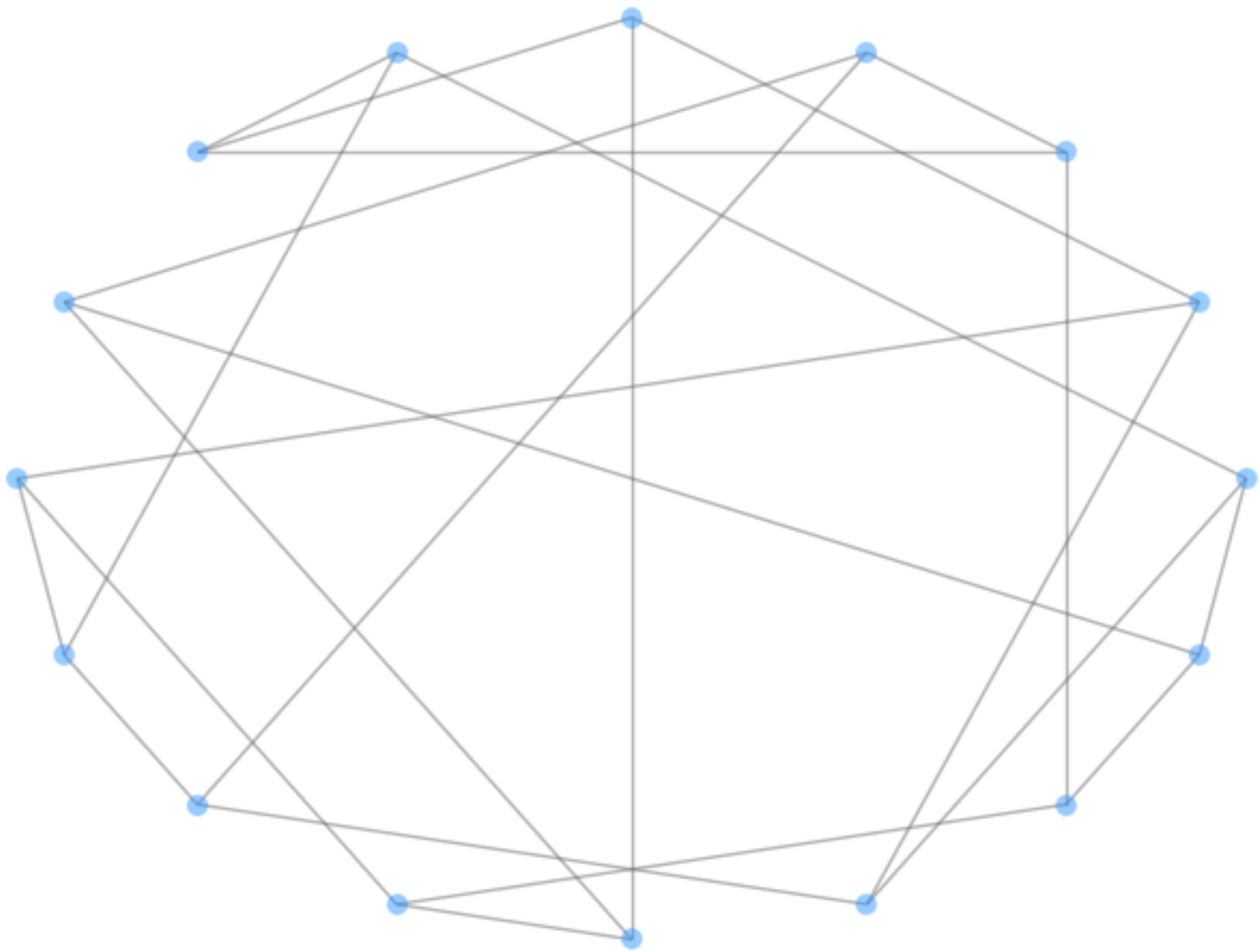


図 : Graph Golf 2015のページより

グラフ例 : $n = 16$, $d = 3$; $k = 3$, $l = 2.200$

(作者 : Komine Shunta)

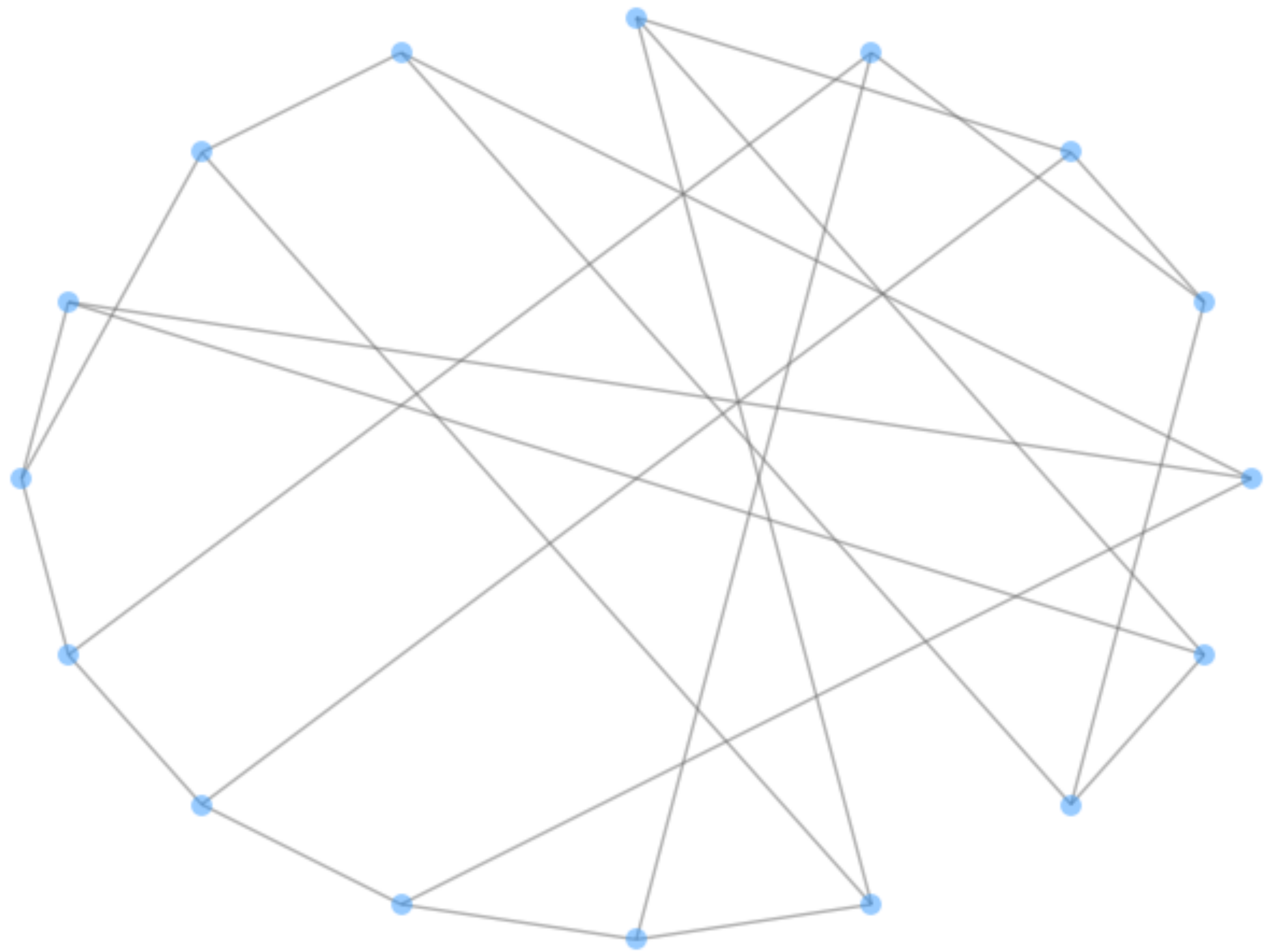


図 : Graph Golf 2015のページより

グラフ例 : n = 16, d = 3; k = 3, l = 2.200
(作者 : Koji Nakano)

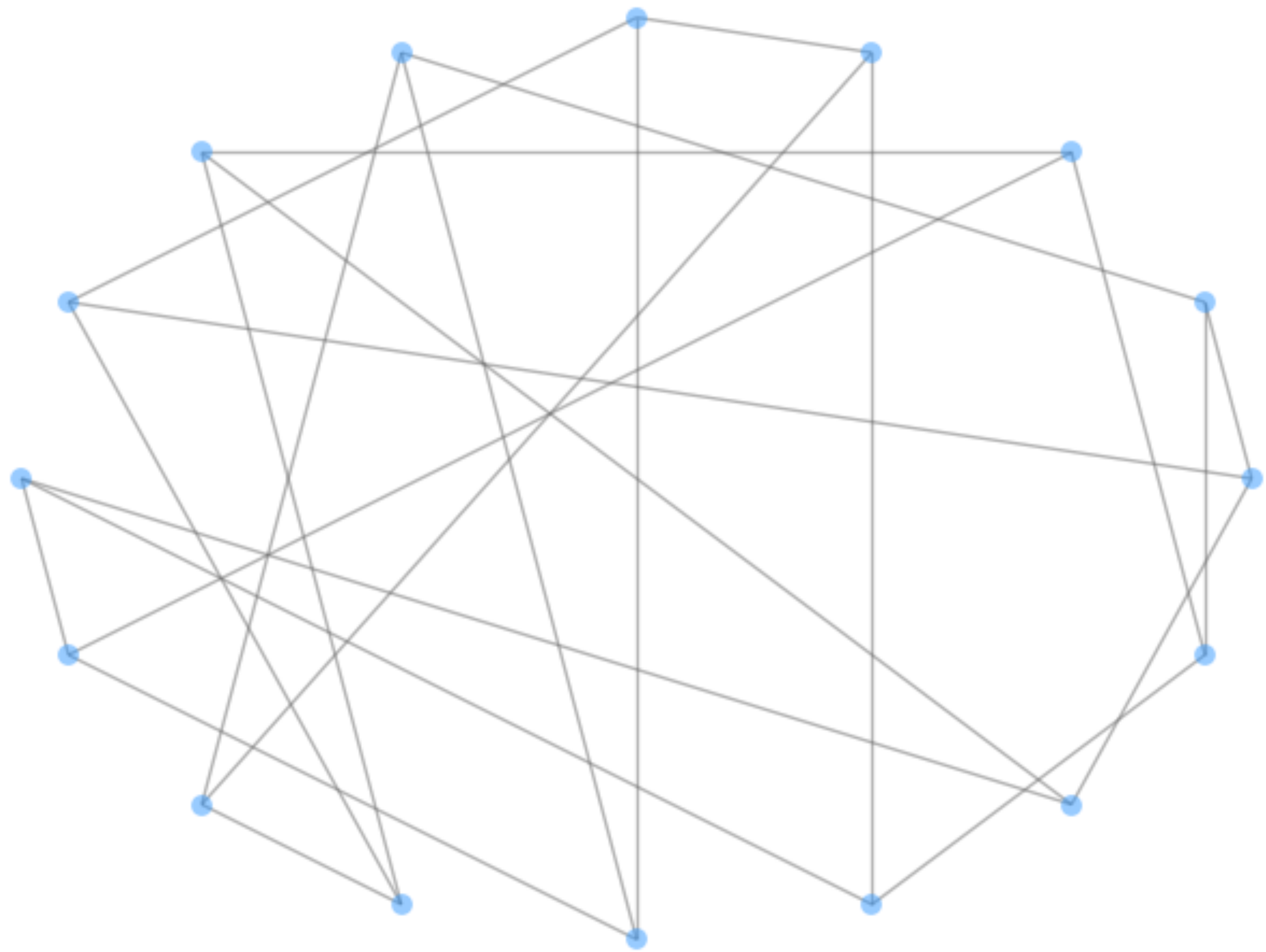


図 : Graph Golf 2015のページより

グラフ例 : $n = 16,\ d = 3;\ k = 3,\ l = 2.200$

(作者 : imazato)

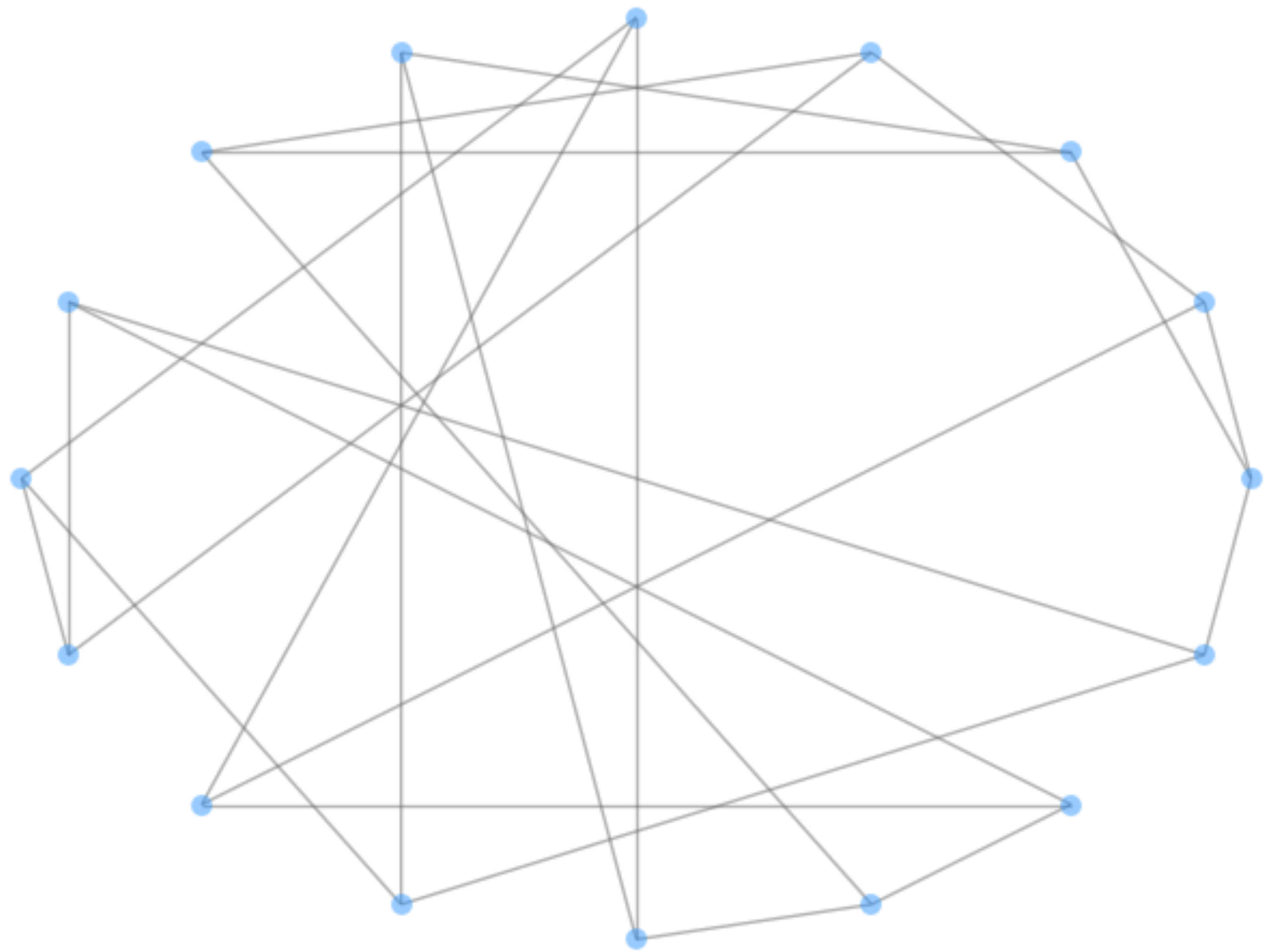


図 : Graph Golf 2015のページより

グラフ例 : n = 16, d = 3; k = 3, l = 2.200
(作者 : Teruaki Kitasuka & Masahiro Iida)

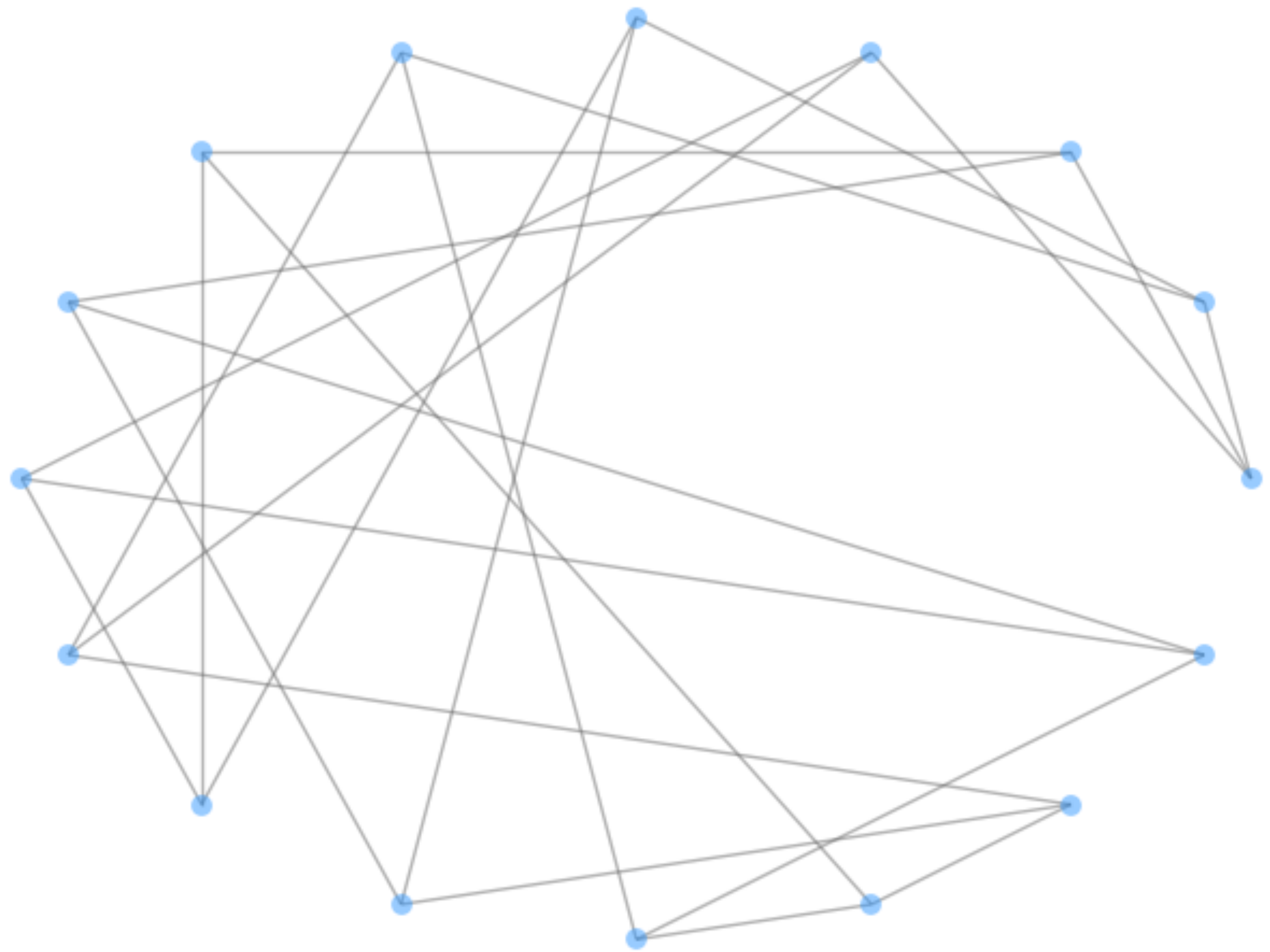


図 : Graph Golf 2015のページより

グラフ例 : n = 16, d = 3; k = 3, l = 2.200

(作者 : yuta,kakibuka)

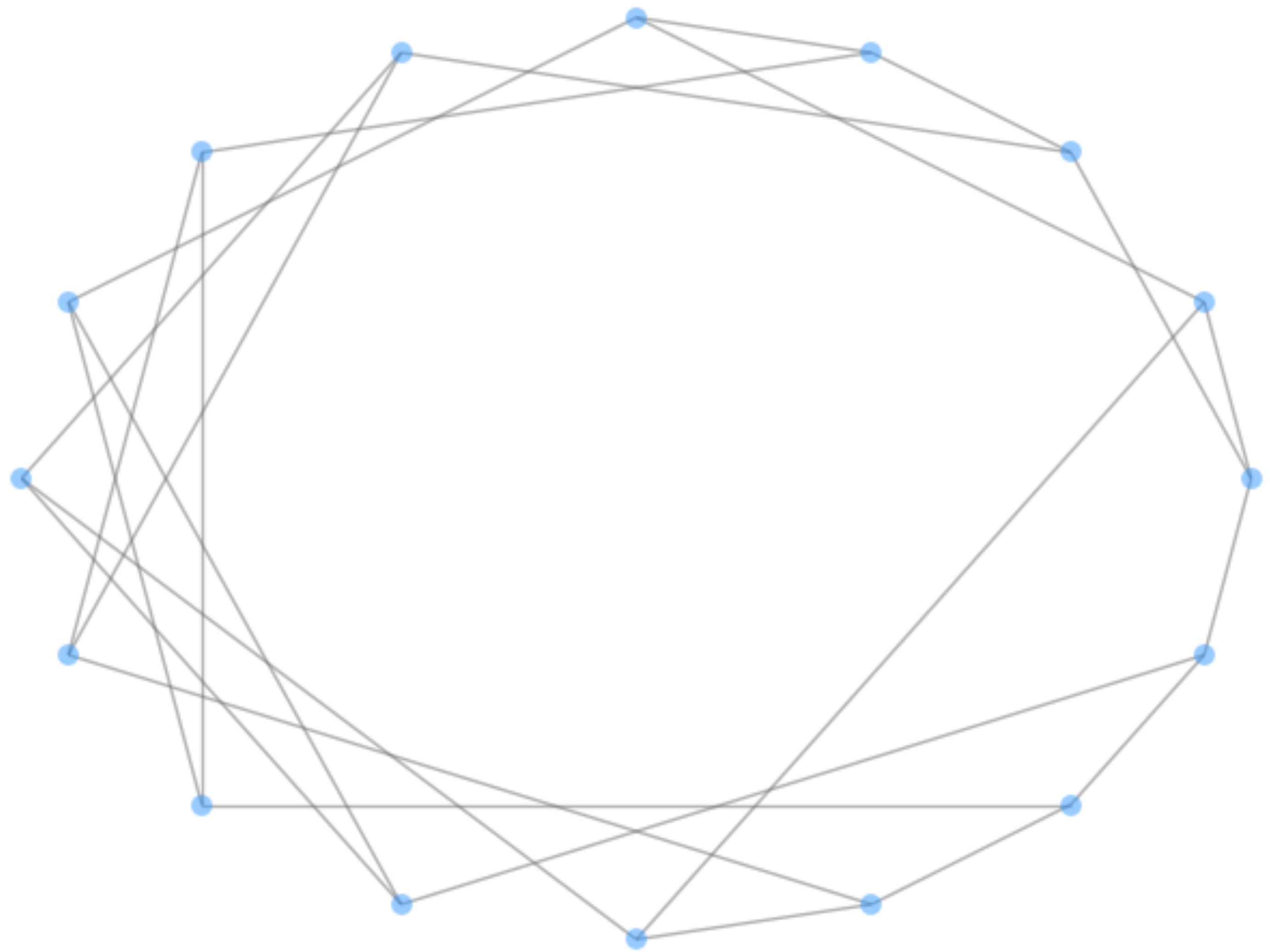


図 : Graph Golf 2015のページより

グラフ例 : $n = 16$, $d = 3$; $k = 3$, $l = 2.200$

(作者 : azapen6)

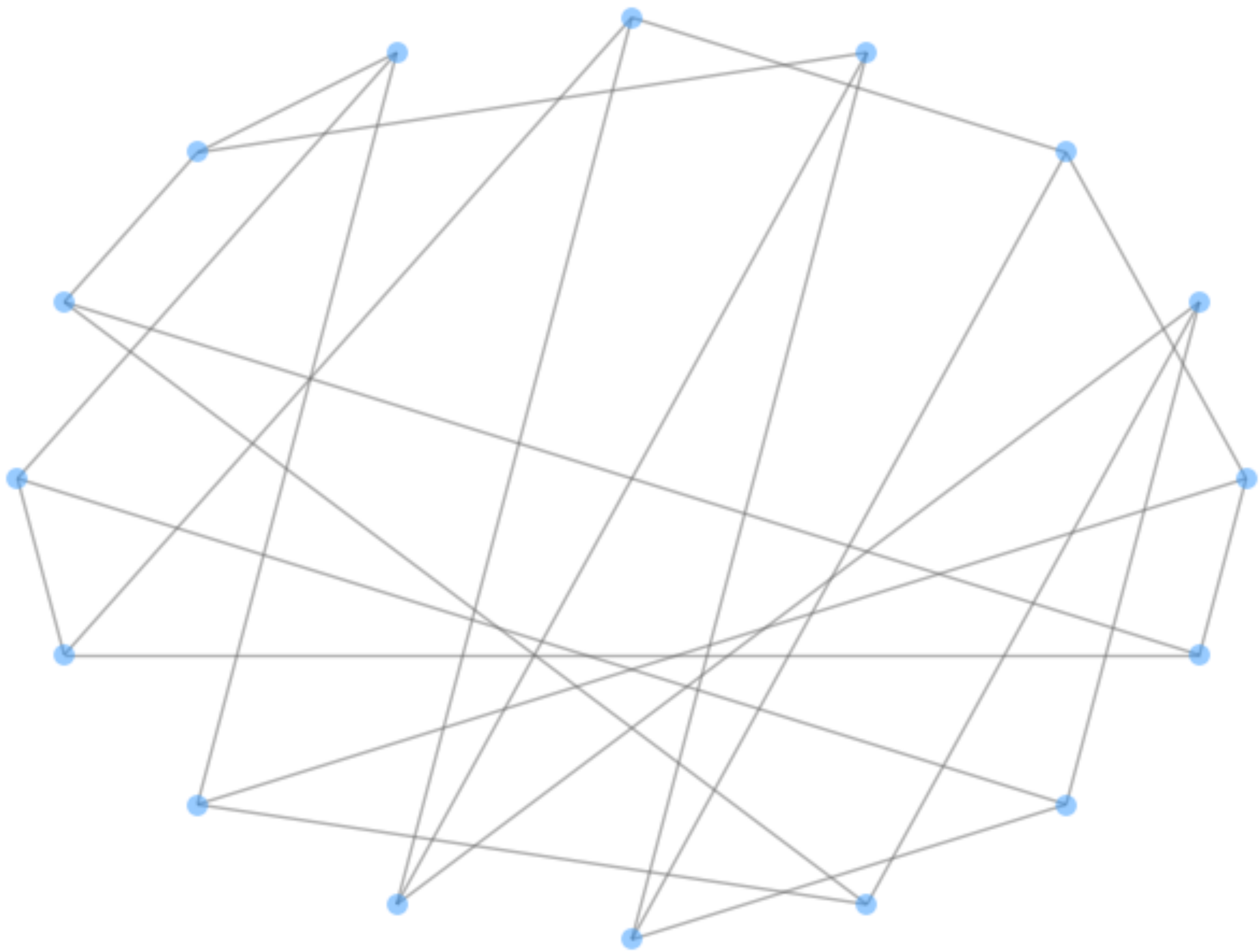
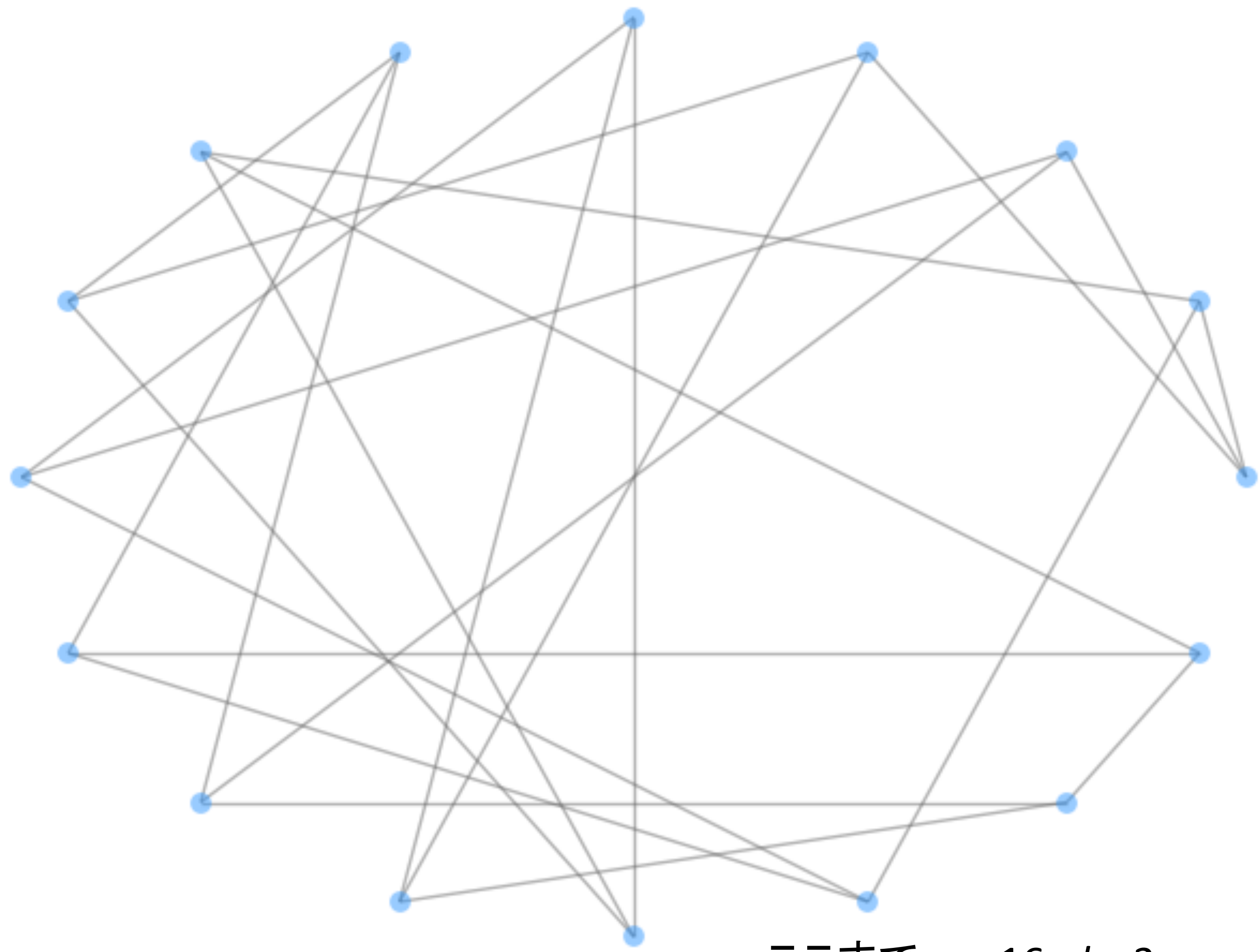


図 : Graph Golf 2015のページより

グラフ例 : n = 16, d = 3; k = 3, l = 2.200
(作者 : Toshihiro Shimizu)



ここまで n = 16, d = 3
図 : Graph Golf 2015のページより

グラフ例 : n = 16, d = 4; k = 3, l = 1.750

(作者 : Teruaki Kitasuka & Masahiro Iida. 次の3つも同じ)

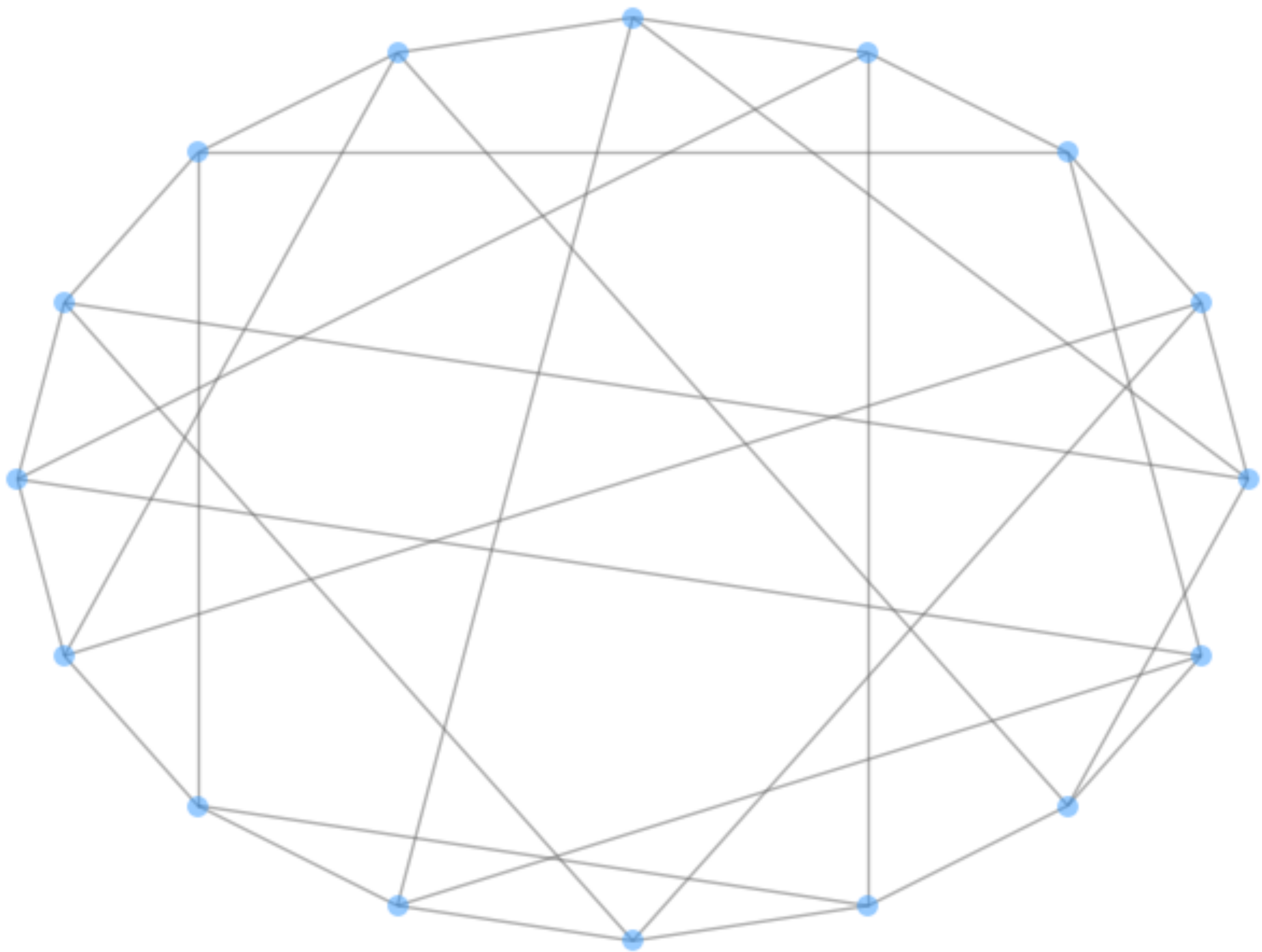


図 : Graph Golf 2015のページより

グラフ例 : n = 256, d = 16; k = 3, l = 2.091

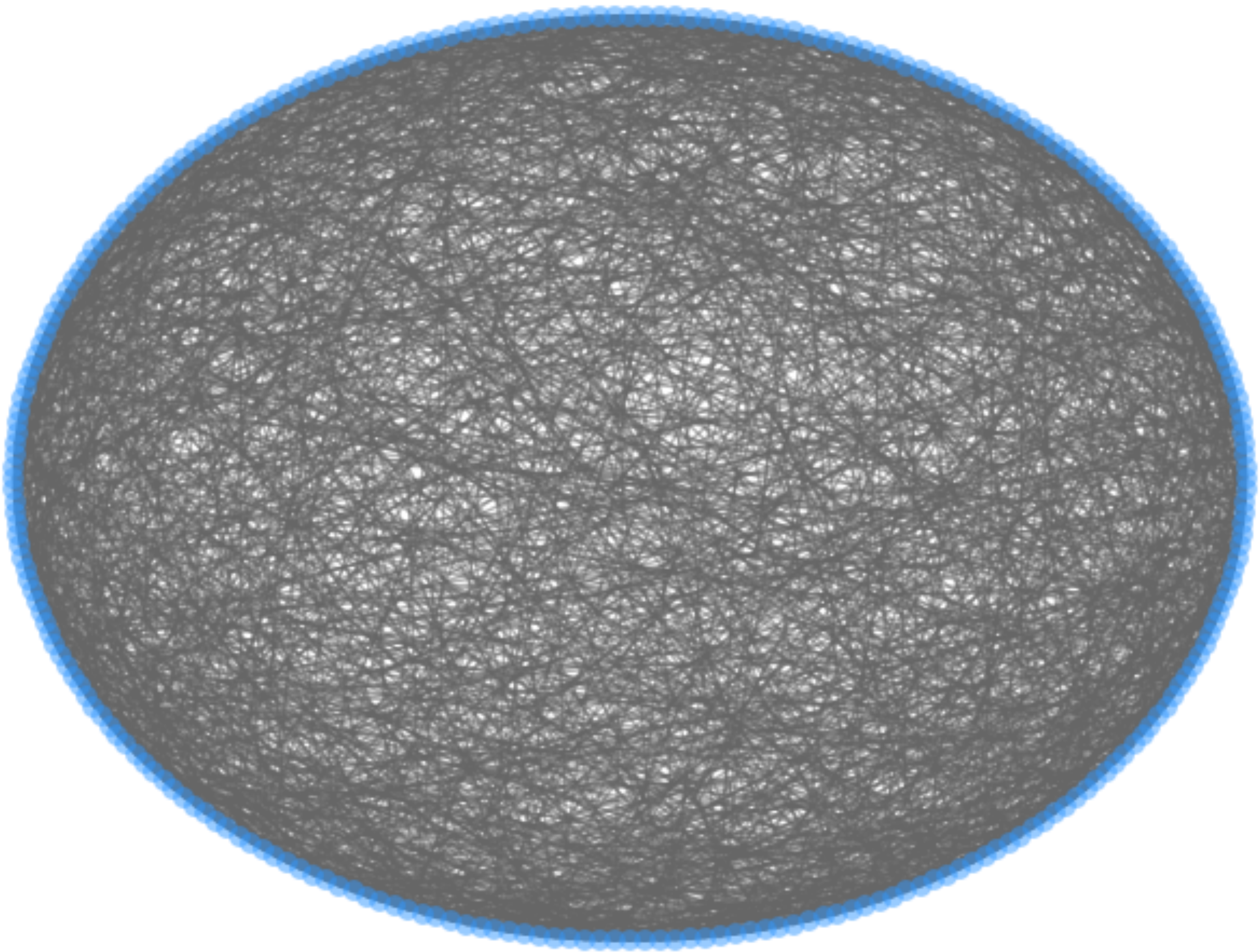


図 : Graph Golf 2015のページより

グラフ例 : $n = 4096,\ d = 60;\ k = 3,\ l = 2.295$

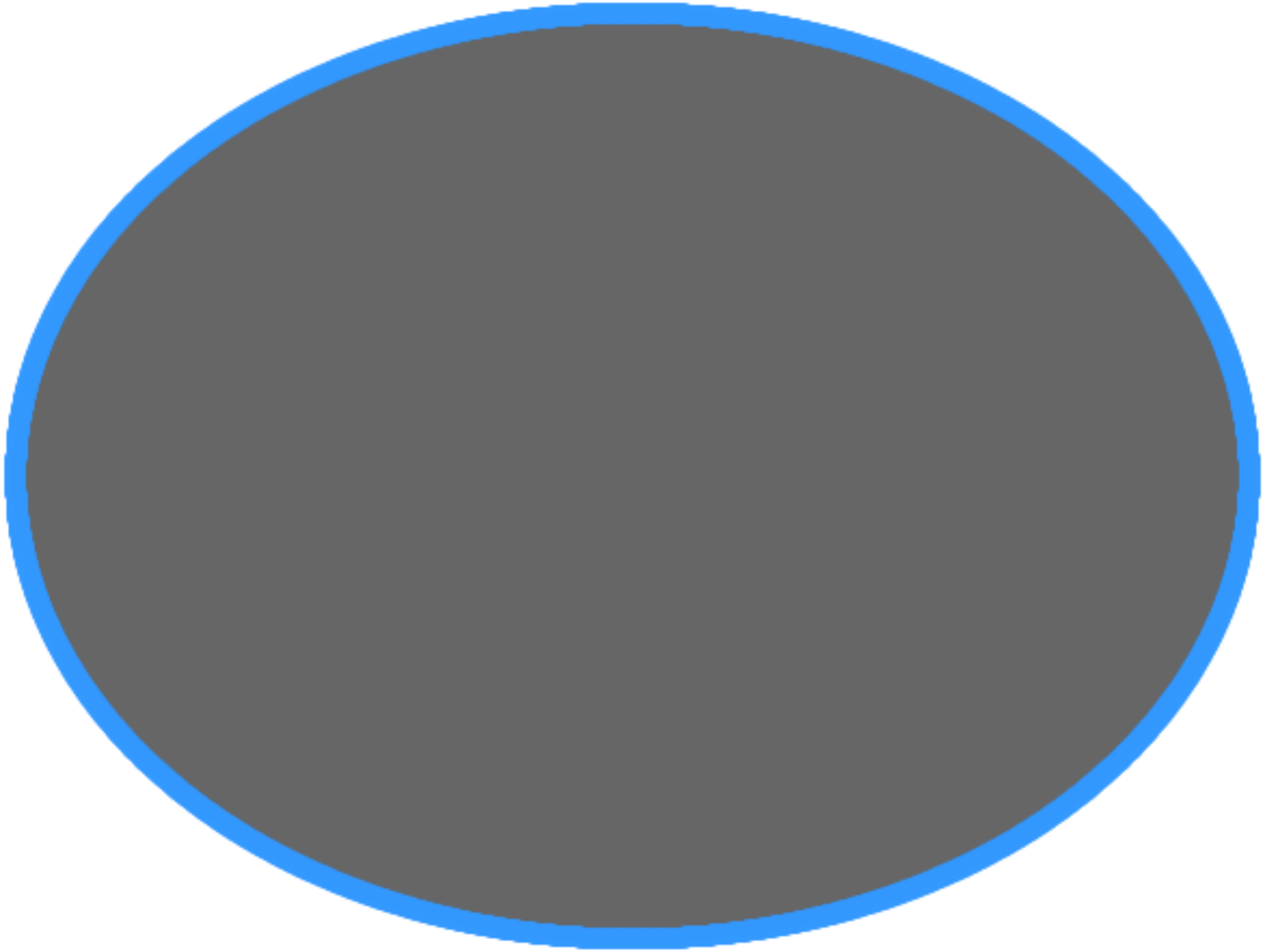


図 : Graph Golf 2015のページより

グラフ例: $n = 4096$, $d = 64$; $k = 3$, $l = 2.242$

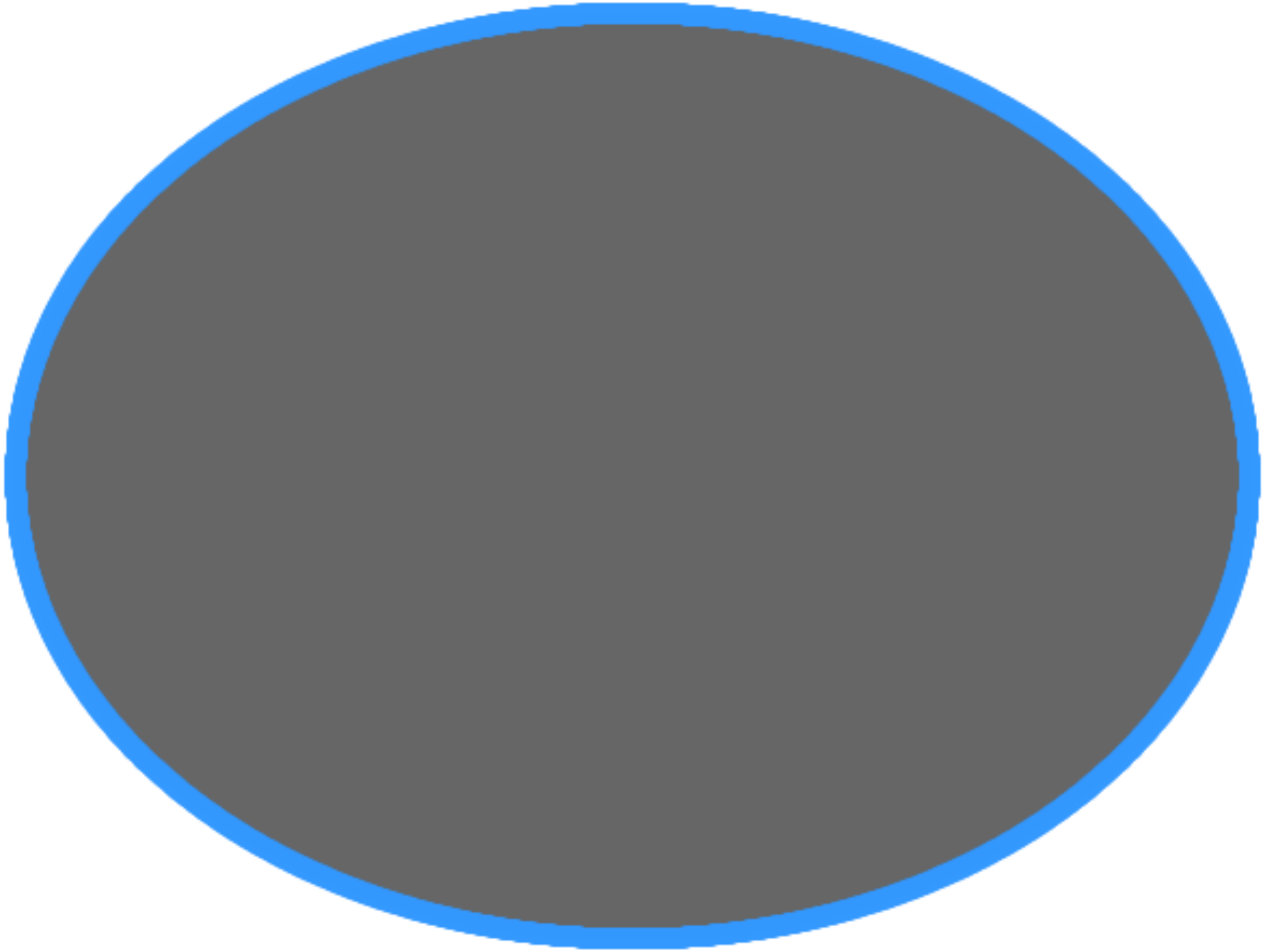


図: Graph Golf 2015のページより

次数・直径問題（類似問題）の既知の解
（与えられた次数と直径に対してノード数を最大化）

d / *k*	2	3	4	5	6	7	8	9	10
3	10	20	38	70	132	196	336	600	1250
4	15	41	98	364	740	1320	3243	7575	17703
5	24	72	212	624	2772	5516	17030	57840	187056
6	32	111	390	1404	7917	19383	76461	331387	1253615
7	50	168	672	2756	11988	52768	249660	1223050	6007230
8	57	253	1100	5060	39672	131137	734820	4243100	24897161

d: 次数（ノードから出る枝の数の最大値）, *k*: 直径（最も遠いノード間の距離）

Petersenグラフと Hoffman–Singletonグラフ.（下界に一致)

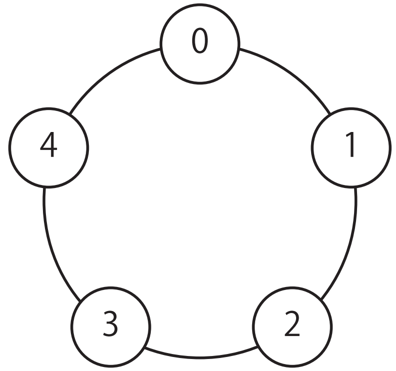
最適と証明されているけど，下界より大きい

Combinatorics Wiki より
http://moorebound.indstate.edu/wiki/The_Degree_Diameter_Problem_for_General_Graphs

次数・直径問題の解の観察

直径 k = 2 の最大グラフは，次数 d によって

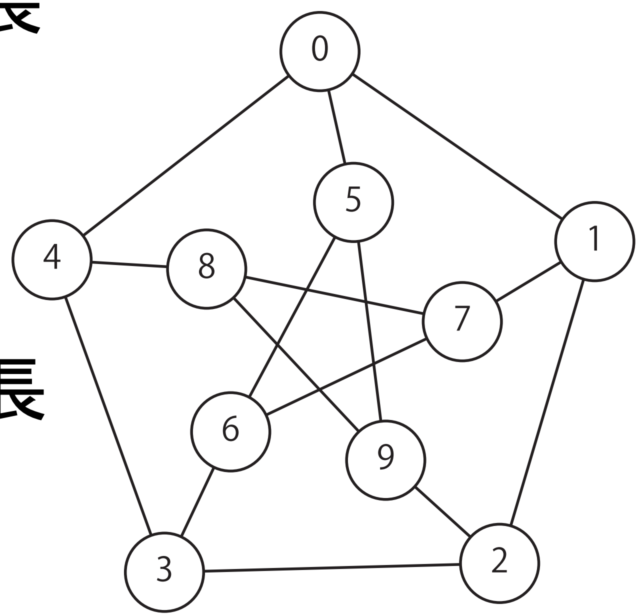
- d = 2: 5ノードの閉路
- d = 3: Petersenグラフ (n = 10)
- d = 7: Hoffman-Singleton グラフ (n = 50)
- d > 3: 4ノード以下の閉路はおそらく冗長



直径 k = 3 のグラフは，次数 d によって

- d = 2: 7ノードの閉路
- d > 2: 6ノード以下の閉路はおそらく冗長

7ノードの閉路を作ろうかと思ったが．．．



手書きグラフの観察

$n = 16,\ d = 3;\ k = 3,\ l = 2.200$ の12個のグラフ

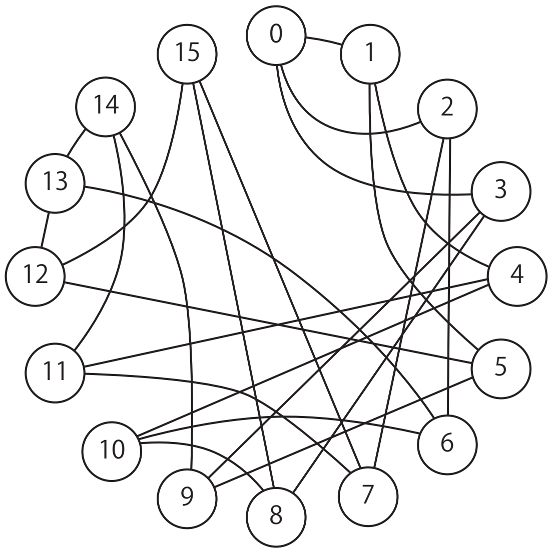
- 先の12個のグラフは，同型のものをまとめると実は5種類らしい [NetworkX (a Python package), is_isomorphic]
- 番号のつけ方が違うだけのグラフを同型のグラフという

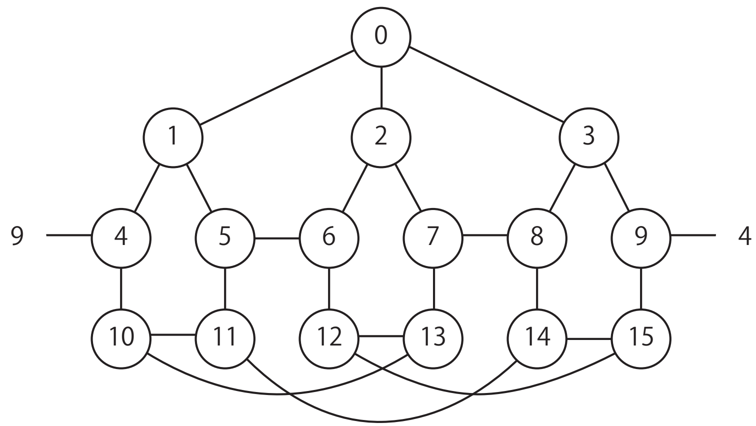
Nobushimi & Ryo Ashida & Ryuhei Mori	H. Inoue	yawara & amami Komine Shunta	imazato	Toshihiro Shimizu
flowlight	Teruaki Kitasuka & Masahiro Iida	Koji Nakano azapen6	yuta,kakibuka	

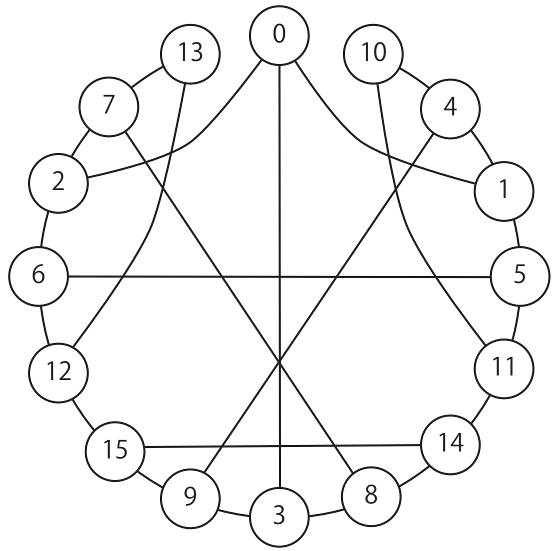
図 : Graph Golf 2015のページより

手書きグラフの観察

n = 16, d = 3; k = 3, l = 2.200



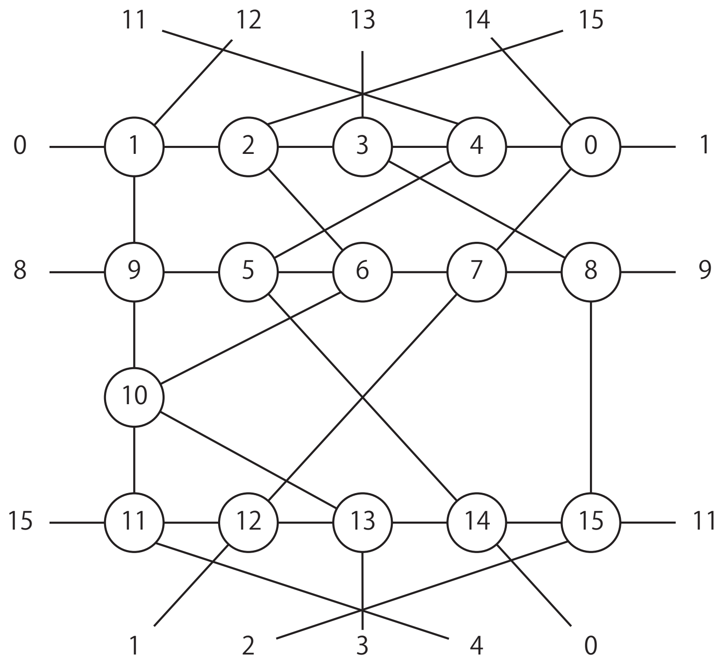
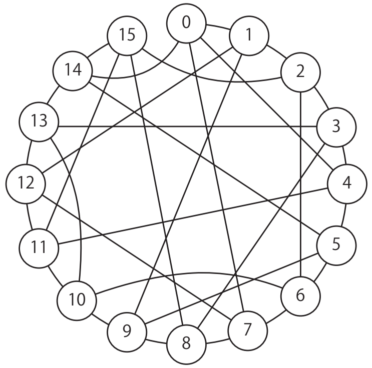
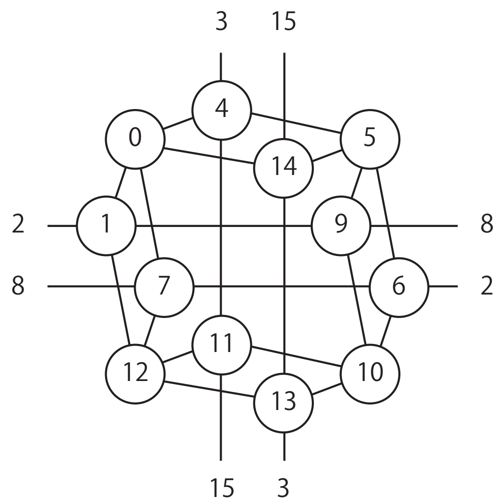
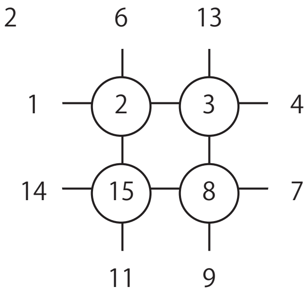




- 5ノードの閉路（5角形）がたくさんあって，4角形なし
- 5ノードの閉路（5角形）をたくさん作ろう！

手書きグラフの観察

n = 16, d = 4; k = 3, l = 1.750

- 5角形がやっぱり多い.
- 4角形は5つ (5-6-10-9, 10-11-12-13, 0-4-5-14, 0-1-12-7, 2-3-8-15)

グラフの作り方：$n = 16$, $d = 4$; $k = 3$, $l = 1.750$ の場合

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0															
1		0														
2			0													
3				0												
4					0											
5						0										
6							0									
7								0								
8									0							
9										0						
10											0					
11												0				
12													0			
13														0		
14															0	
15																0

グラフの作り方：
n = 16, d = 4; k = 3, l = 1.750 の場合

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	1	2	3	4											
1	1	0	1	2	3	4										
2	2	1	0	1	2	3	4									
3	3	2	1	0	1	2	3	4								
4	4	3	2	1	0	1	2	3	4							
5		4	3	2	1	0	1	2	3	4						
6			4	3	2	1	0	1	2	3	4					
7				4	3	2	1	0	1	2	3	4				
8					4	3	2	1	0	1	2	3	4			
9						4	3	2	1	0	1	2	3	4		
10							4	3	2	1	0	1	2	3	4	
11								4	3	2	1	0	1	2	3	4
12									4	3	2	1	0	1	2	3
13										4	3	2	1	0	1	2
14											4	3	2	1	0	1
15												4	3	2	1	0

距離5以上は省略

グラフの作り方：
$n = 16$, $d = 4$; $k = 3$, $l = 1.750$ の場合

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
0	0	1	2	2	1	2	3	4								
1	1	0	1	2	2	3	4									
2	2	1	0	1	2	3	4									
3	2	2	1	0	1	2	3	4								
4	1	2	2	1	0	1	2	3	4							
5	2	3	3	2	1	0	1	2	3	4						
6	3	4	4	3	2	1	0	1	2	3	4					
7	4			4	3	2	1	0	1	2	3	4				
8					4	3	2	1	0	1	2	3	4			
9						4	3	2	1	0	1	2	3	4		
10							4	3	2	1	0	1	2	3	4	
11								4	3	2	1	0	1	2	3	4
12									4	3	2	1	0	1	2	3
13										4	3	2	1	0	1	2
14											4	3	2	1	0	1
15												4	3	2	1	0

距離5以上は省略

グラフの作り方：
n = 16, d = 4; k = 3, l = 1.750 の場合

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
0	0	1	2	2	1	2	2	1	2	3	4					
1	1	0	1	2	2	3	3	2	3	4						
2	2	1	0	1	2	3	4	3	4							
3	2	2	1	0	1	2	3	3	4							
4	1	2	2	1	0	1	2	2	3	4						
5	2	3	3	2	1	0	1	2	3	4						
6	2	3	4	3	2	1	0	1	2	3	4					
7	1	2	3	3	2	2	1	0	1	2	3	4				
8	2	3	4	4	3	3	2	1	0	1	2	3	4			
9	3	4			4	4	3	2	1	0	1	2	3	4		
10	4						4	3	2	1	0	1	2	3	4	
11								4	3	2	1	0	1	2	3	4
12									4	3	2	1	0	1	2	3
13										4	3	2	1	0	1	2
14											4	3	2	1	0	1
15												4	3	2	1	0

距離5以上は省略

グラフの作り方：
n = 16, d = 4; k = 3, l = 1.750 の場合

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
0	0	1	2	2	1	2	2	1	2	3	4					
1	1	0	1	2	2	3	2	2	3	4						
2	2	1	0	1	2	2	1	2	3	4						
3	2	2	1	0	1	2	2	3	4							
4	1	2	2	1	0	1	2	2	3	4						
5	2	3	2	2	1	0	1	2	3	4						
6	2	2	1	2	2	1	0	1	2	3	4					
7	1	2	2	3	2	2	1	0	1	2	3	4				
8	2	3	3	4	3	3	2	1	0	1	2	3	4			
9	3	4	4		4	4	3	2	1	0	1	2	3	4		
10	4						4	3	2	1	0	1	2	3	4	
11								4	3	2	1	0	1	2	3	4
12									4	3	2	1	0	1	2	3
13										4	3	2	1	0	1	2
14											4	3	2	1	0	1
15												4	3	2	1	0

距離5以上は省略

グラフの作り方：
$n = 16$, $d = 4$; $k = 3$, $l = 1.750$ の場合

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	1	2	2	1	2	2	1	2	3	4					
1	1	0	1	2	2	3	2	2	3	4						
2	2	1	0	1	2	2	1	2	2	3	4					
3	2	2	1	0	1	2	2	2	1	2	3	4				
4	1	2	2	1	0	1	2	2	2	3	4					
5	2	3	2	2	1	0	1	2	3	4						
6	2	2	1	2	2	1	0	1	2	3	4					
7	1	2	2	2	2	2	1	0	1	2	3	4				
8	2	3	2	1	2	3	2	1	0	1	2	3	4			
9	3	4	3	2	3	4	3	2	1	0	1	2	3	4		
10	4		4	3	4		4	3	2	1	0	1	2	3	4	
11				4				4	3	2	1	0	1	2	3	4
12									4	3	2	1	0	1	2	3
13										4	3	2	1	0	1	2
14											4	3	2	1	0	1
15												4	3	2	1	0

グラフの作り方：
n = 16, d = 4; k = 3, l = 1.750 の場合

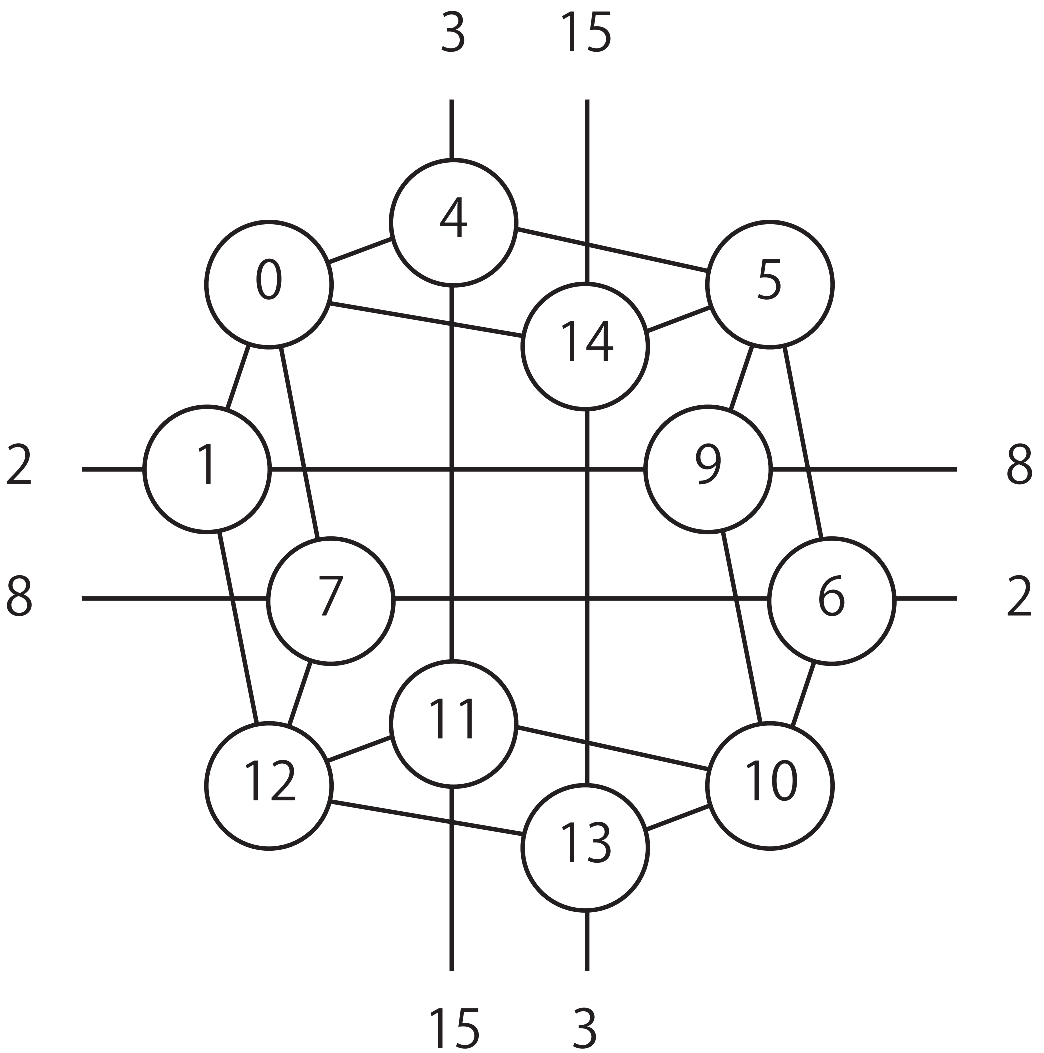
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
0	0	1	2	2	1	2	2	1	2	2	3	4				
1	1	0	1	2	2	3	2	2	2	1	2	3	4			
2	2	1	0	1	2	2	1	2	2	3	3	4				
3	2	2	1	0	1	2	2	2	1	2	3	4				
4	1	2	2	1	0	1	2	2	2	3	4					
5	2	3	2	2	1	0	1	2	3	4						
6	2	2	1	2	2	1	0	1	2	3	4					
7	1	2	2	2	2	2	1	0	1	2	3	4				
8	2	2	2	1	2	3	2	1	0	1	2	3	4			
9	2	1	2	2	3	4	3	2	1	0	1	2	3	4		
10	3	2	3	3	4		4	3	2	1	0	1	2	3	4	
11	4	3	4	4				4	3	2	1	0	1	2	3	4
12		4							4	3	2	1	0	1	2	3
13										4	3	2	1	0	1	2
14											4	3	2	1	0	1
15												4	3	2	1	0

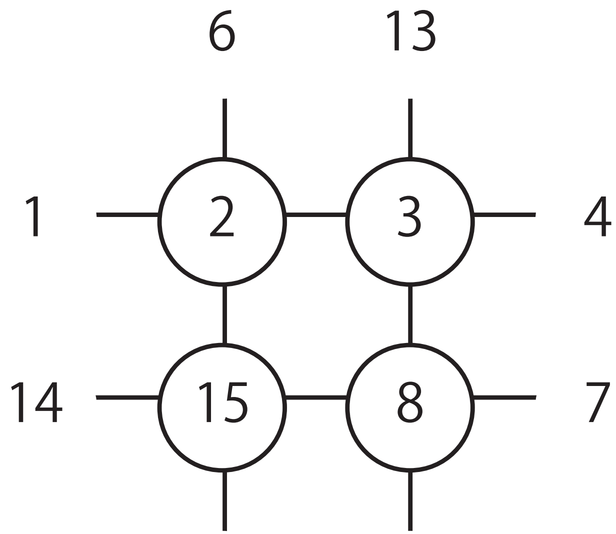
グラフの作り方：
$n = 16,\ d = 4;\ k = 3,\ l = 1.750$ の場合

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	1	2	2	1	2	2	1	2	2	3	2	2	2	1	2
1	1	0	1	2	2	2	2	2	2	1	2	2	1	2	2	2
2	2	1	0	1	2	2	1	2	2	2	2	2	2	2	2	1
3	2	2	1	0	1	2	2	2	1	2	2	2	2	1	2	2
4	1	2	2	1	0	1	2	2	2	2	2	1	2	2	2	2
5	2	2	2	2	1	0	1	2	2	1	2	2	3	2	1	2
6	2	2	1	2	2	1	0	1	2	2	1	2	2	2	2	2
7	1	2	2	2	2	2	1	0	1	2	2	2	1	2	2	2
8	2	2	2	1	2	2	2	1	0	1	2	2	2	2	2	1
9	2	1	2	2	2	1	2	2	1	0	1	2	2	2	2	2
10	3	2	2	2	2	2	1	2	2	1	0	1	2	1	2	2
11	2	2	2	2	1	2	2	2	2	2	1	0	1	2	2	1
12	2	1	2	2	2	3	2	1	2	2	2	1	0	1	2	2
13	2	2	2	1	2	2	2	2	2	2	3	2	1	0	1	2
14	1	2	2	2	2	1	2	2	2	2	2	2	2	1	0	1
15	2	2	1	2	2	2	2	2	1	2	2	4	2	2	1	0

手書きグラフの観察

$n = 16,\ d = 4;\ k = 3,\ l = 1.750$（図再掲）





経験的手法による直径3のグラフ生成

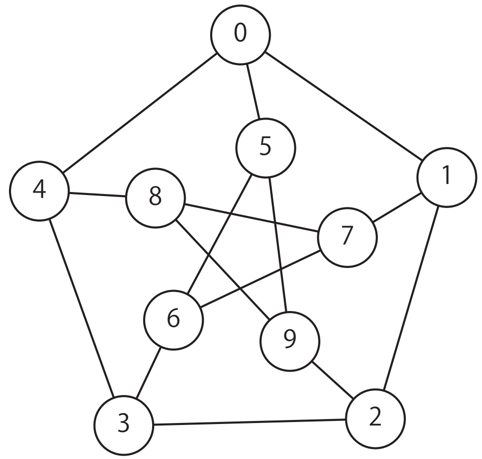
対象グラフ

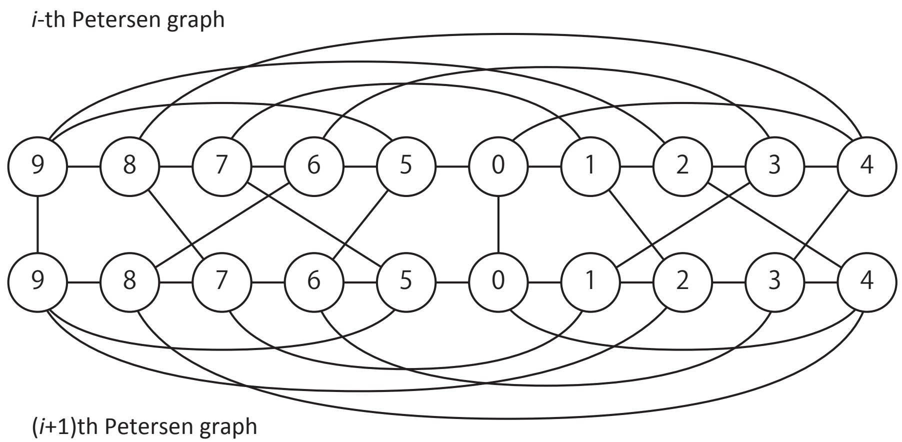
経験的手法が有効

degree d	Order n		
	256	**4096**	**10000**
16	(1.93725 下界．直径2) 2.09069 (コンペ後) (2.09262 コンペ最良) <u>2.12757</u> (2.28186 random)		
60		(2.106227 下界) 2.295216 (コンペ後) <u>2.295275</u> (Aug.) (2.395112 random)	(2.633963 下界) 2.648977 (コンペ後) <u>2.648980</u> (コンペ後) (2.650399 コンペ最良)
64		(1.984371 下界．直径2) 2.242170 (コンペ後) <u>2.242228</u> (Sep.,) (2.346357 random)	(2.583958 下界) (2.610117 コンペ最良) <u>2.611310</u> (コンペ後) (2.655734 random)

経験的手法による直径3のグラフ生成
あらまし

1. 与えられたノード数になるように，複数のPetersen グラフ（直径2）をつないで基本グラフ G_0 を作る．次数は5
2. 与えられた次数をこえないように，貪欲（Greedy）に枝を1本ずつ追加する．5角形の数が増えるように欲張る

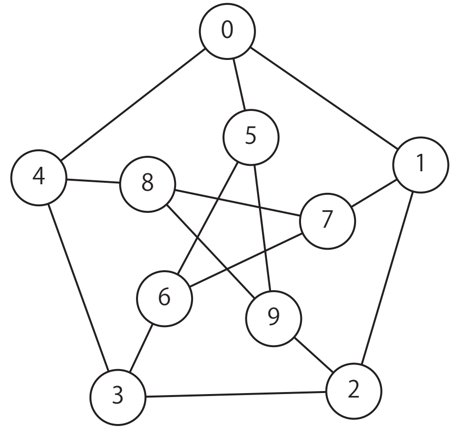


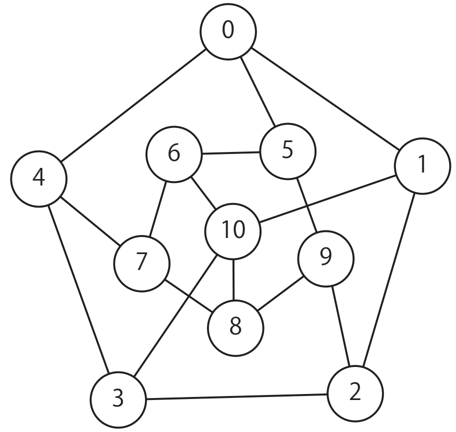


経験的手法による直径3のグラフ生成

1: 基本グラフ G_0 の生成　(1/2)

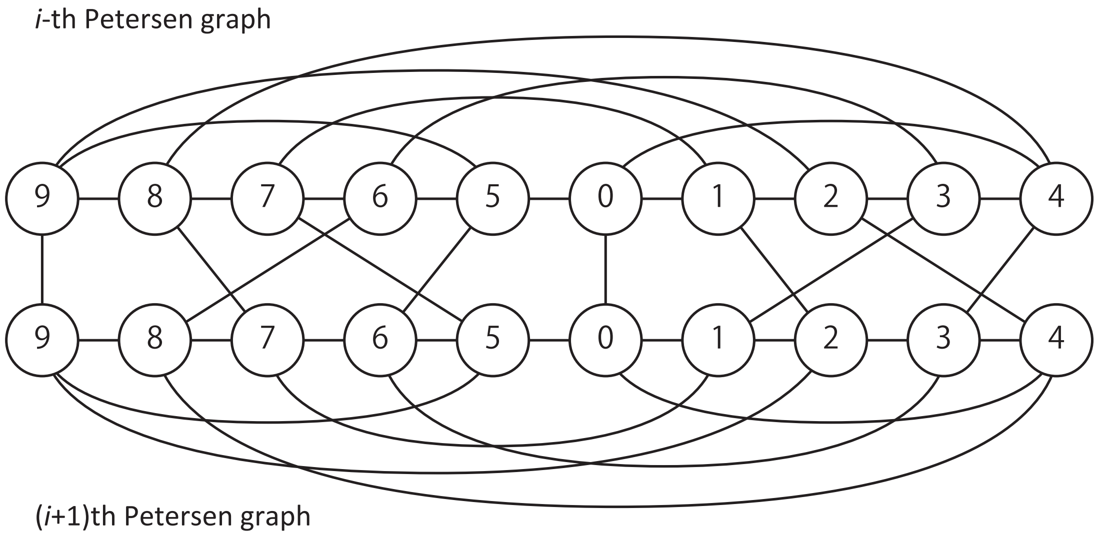
- Petersen グラフ（直径は2）を用意つないで，与えられたノード数のグラフを作る
（直径4のグラフを作るとしたら，Petersen グラフの代わりに直径3のグラフを使うとよいだろう）
- n = 10000: 1000個の Petersen グラフをつなぐ
- n= 4096: (409 – 6)個の Petersen グラフと6個の11ノードグラフ（右下図）をつなぐ





経験的手法による直径3のグラフ生成
1: 基本グラフ G_0 の生成 (2/2)

- ノード数 n のとき，m = $\lfloor n/10 \rfloor$ 個のPetersenグラフ（一部は11ノードグラフ）
- Petersenグラフを一列に並べ，隣り合う2つのグラフの各ノード同士を右図のようにつなぐ
- つなぎ終わると，次数は5になる．一部のノードの次数は4



経験的手法による直径3のグラフ生成
2 : G_0への貪欲（Greedy）な枝の追加 (1/2)

枝を追加するときの方針

- 新しい枝の一端のノード i は，次数が最小のノードから選ぶ
- 新しい枝の他端のノード j は，5角形の数が増えるように<u>貪欲に</u>選ぶ
- 一度追加した枝は削除しない

Algorithm 3 Greedily add edges, one by one to G_0

```
procedure AddEdges(n, d, G_0)
    G ← G_0
    while edge can be added do
        Select a node i from the smallest degree nodes
        Compute node set J such that d(i, j) > 2
        for each node j ∈ J do
            p_1(j) = CountPaths(i, j)
            p_2(j) = 0
            for each k ∈ j's neighbors do
                if CountPaths(i, k) > p_2(j) then
                    p_2(j) = CountPaths(i, k)
                end if
            end for
        end for
        Select j ∈ J that satisfies conditions (1) and (2)
        Add an edge i-j to graph G
    end while
    If degree of several nodes are less than d, add several
    edges between them
end procedure
function CountPaths(i, j)
    p = |D_1(i) ∩ D_2(j)| + |D_2(i) ∩ D_1(j)|    ▷ Roughly
    count the number of paths with distance 3 between node
    i and j, for graph G
end function
```

経験的手法による直径3のグラフ生成

2 : G_0への貪欲（Greedy）な枝の追加 (2/2)

5角形の数が増える *j* の探し方

- CountPaths 関数で，指定した2ノード間にある長さ3の道の数を数える
- 枝の候補 (*i*, *j*) と *j* に隣接するノード *k* に対して，CountPaths(*i*, *k*) を計算し，CountPathsの値が最大になる *k* に対応する *j* を選び．枝 (*i*, *j*) を追加する

Algorithm 3 Greedily add edges, one by one to G_0

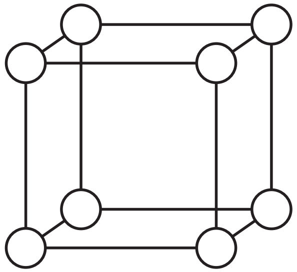
procedure AddEdges(n, d, G_0)
　$G \leftarrow G_0$
　while edge can be added **do**
　　Select a node i from the smallest degree nodes
　　Compute node set J such that $d(i, j) > 2$
　　for each node $j \in J$ **do**
　　　$p_1(j) = \text{CountPaths}(i, j)$
　　　$p_2(j) = 0$
　　　for each $k \in j$'s neighbors **do**
　　　　if $\text{CountPaths}(i, k) > p_2(j)$ **then**
　　　　　$p_2(j) = \text{CountPaths}(i, k)$
　　　　end if
　　　end for
　　end for
　　Select $j \in J$ that satisfies conditions (1) and (2)
　　Add an edge i-j to graph G
　end while
　If degree of several nodes are less than d, add several edges between them
end procedure
function CountPaths(i, j)
　$p = |D_1(i) \cap D_2(j)| + |D_2(i) \cap D_1(j)|$ ▷ Roughly count the number of paths with distance 3 between node i and j, for graph G
end function

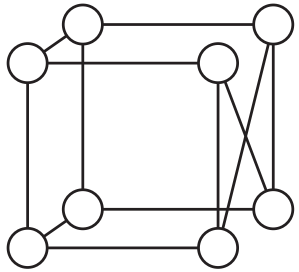
経験的手法による直径3のグラフ生成
対象グラフ

経験的手法が有効

degree *d*	Order *n*		
	256	**4096**	**10000**
16	(1.93725 下界. 直径2) 2.09069 (2-OPT) (2.09262 コンペ最良) <u>2.12757</u> (経験的) (2.28186 random)		
60		(2.106227 下界) 2.295216 (2-OPT) <u>2.295275</u> (経験的) (2.395112 random)	(2.633963 下界) 2.648977 (2-OPT) <u>2.648980</u> (経験的) (2.650399 コンペ最良)
64		(1.984371 下界. 直径2) 2.242170 (2-OPT) <u>2.242228</u> (経験的) (2.346357 random)	(2.583958 下界) (2.610117 コンペ最良) <u>2.611310</u> (経験的) (2.655734 random)

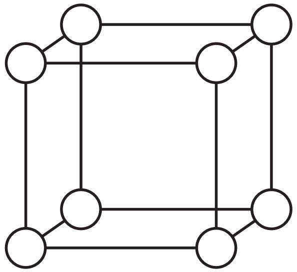
2-OPT局所探索とは2枝のつなぎかえ


Cube; k=3, l=1.71 (Gap 9.1%)

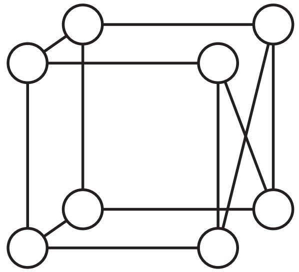

Möbius loop; k=2, l=1.57 (Gap 0.0%)

- 2-Opt:二つ枝をつなぎかえることで，直径と平均パス長が縮むならばつなぎかえる
- 二つの枝の組合せはたくさんある.
  枝数 $m = nd/2$. 二つの枝の組合せ数 $m(m\text{-}1)/2$
- 例えばノード数 n = 10,000，次数 d = 60 ならば，
  $m = 6\times10^5$，$m(m\text{-}1)/2$ = 179,999,700,000 ≒ 1.8×10^{11}

2-OPT局所探索とは2枝のつなぎかえ



Cube; k=3, l=1.71 (Gap 9.1%)

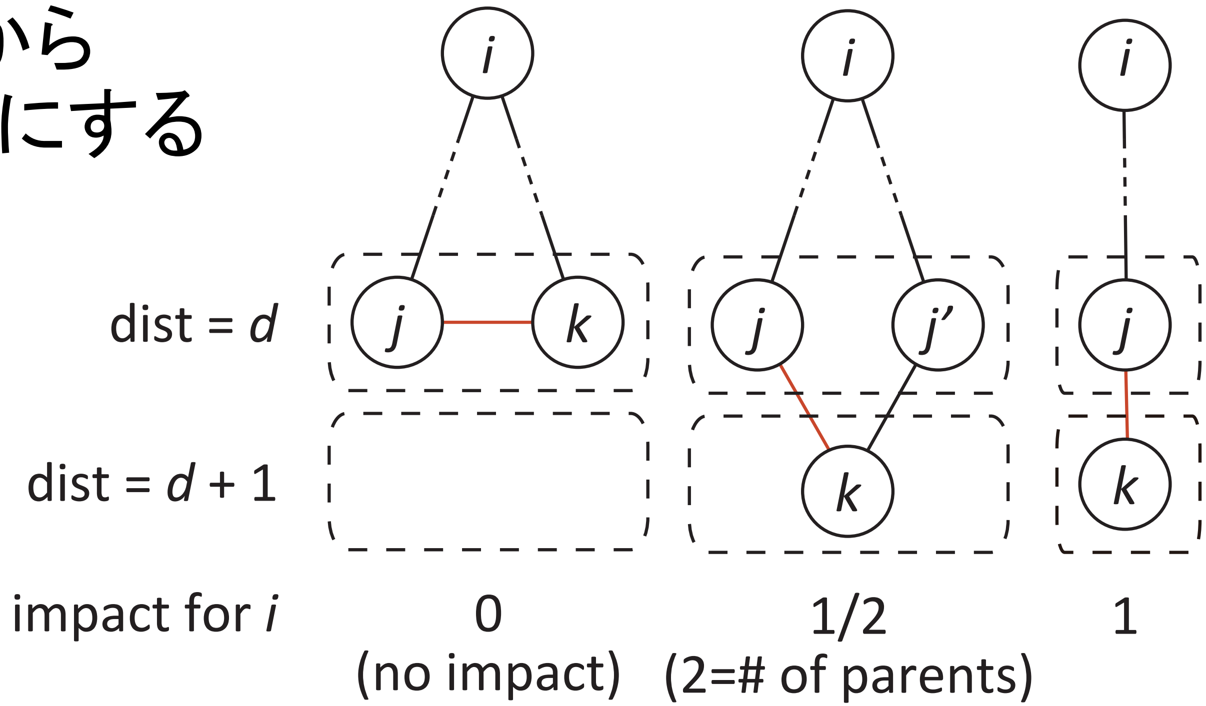


Möbius loop; k=2, l=1.57 (Gap 0.0%)

- 探索空間はとても広い
- 疑問:どの二つの枝をつなぎかえるのがいいのか？
- 一つの解（提案）:重要度の低い枝を優先する

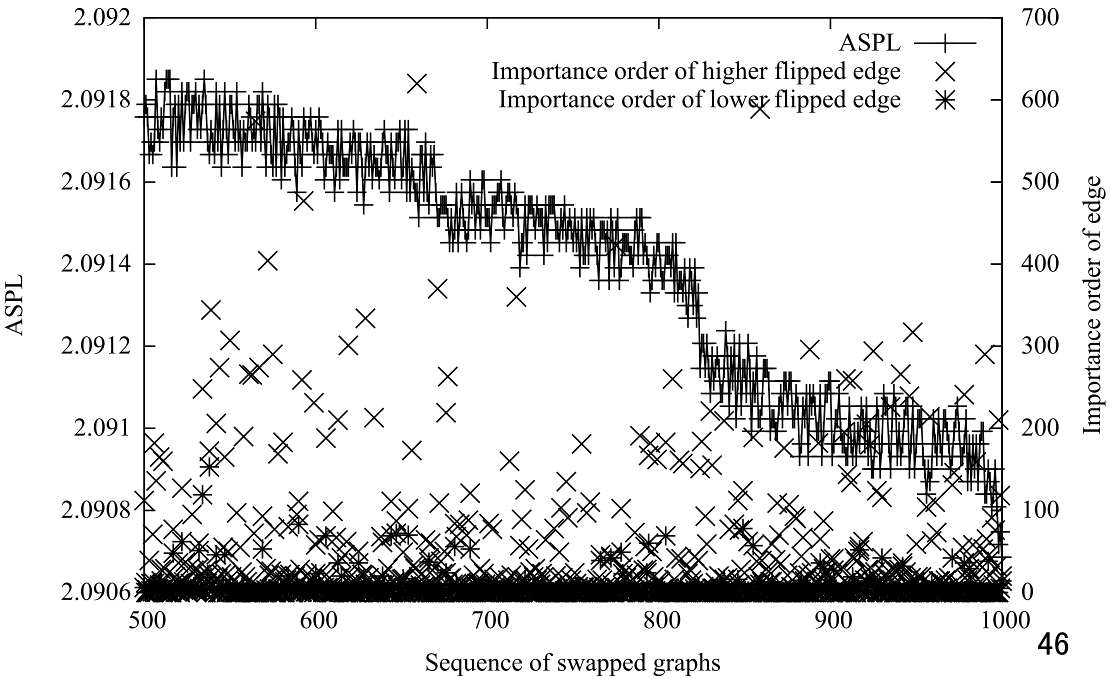
高速2-OPTのための枝の重要度

- （荒っぽい）定義：ノード *i* にとっての枝 (*j*, *k*) が重要なのは，ノード *i* から *j* と *k* までの距離が異なるときである．枝 (*j*, *k*) の重要度はグラフのすべてのノード *i* にとっての重要さ足し合わせたものとする
- 重要度の低い枝から順に2-OPTの対象にする



枝の重要度の使用例

- n = 256, d = 16のグラフ.
  枝数 m = 2048, 二つの枝の組合せ数 2.1×10^6
- 平均パス長 2.09258 から 2.09069が縮む過程の 1000 のグラフの系列を見る. この系列上の二つの枝の組合せ (i, j) は, 重要度順で $0 \le i \le 153$ と $1 \le j \le 620$.
  この範囲は 2.1×10^6 の組合せのうち 4.6% に過ぎない
  (=154 * 621 / 2.1×10^6)
- つまり, 枝の重要度は役に立ちそうである



2-OPT局所探索の技法

対象のグラフ

局所探索で改良

degree d	Order n		
	256	**4096**	**10000**
16	(1.93725 下界．直径2) 2.09069 (2-OPT) (2.09262 コンペ最良) 2.12757 (経験的) (2.28186 random)		
60		(2.106227 下界) 2.295216 (2-OPT) 2.295275 (経験的) (2.395112 random)	(2.633963 下界) 2.648977 (2-OPT) 2.648980 (経験的) (2.650399 コンペ最良)
64		(1.984371 下界．直径2) 2.242170 (2-OPT) 2.242228 (経験的) (2.346357 random)	(2.583958 下界) (2.610117 コンペ最良) 2.611310 (経験的) (2.655734 random)

まとめ

- 直径の小さいグラフの作り方を述べた
- 直径3のグラフには5角形の数が重要そうなことを観察した
- 経験的なグラフ構成法を説明した．その構成法で n = 4096, d = 60 と 64 の最良グラフと，コンペ後には n = 10000, d = 60 の最良グラフを作れた
- 2-OPT局所探索で優先して試す枝の順位付けに，枝の重要度を使うことを提案した．この重要度で，n = 256, d = 16 の最良グラフをコンペ後に見つけた
- 楽しそうだと思いませんか？

NOCS 2016（先週奈良）の原稿あり