

UNIVERZA V LJUBLJANI
FAKULTETA ZA MATEMATIKO IN FIZIKO

Finančna matematika

Marcel Blagotinšek, Peter Milivojević

**Maximum number of edges in a connected graph with n
vertices and diameter d**

Skupinski projekt

Poročilo

Advisers: doc. dr. Janoš Vidali,
prof. dr. Riste Škrekovski

Ljubljana, 2023

1. NAVODILO NALOGE

A connected graph with diameter d on n vertices with the minimal number of edges will be a tree and henceforth, it will have $n - 1$ edges. It will be harder to answer which graphs on a fixed number of vertices n and fixed diameter d have the maximal number of edges. We want to analyse the structure of such graphs. So, for a fixed number of vertices n and a fixed diameter d , when these two values are small, apply an exhaustive search. Next, for larger n and d , apply some metaheuristic. Try to obtain some specific properties of these graphs. Verify for how large n and d your exhaustive search and your metaheuristic implementations are efficient.

2. OPIS PROBLEMA

Želiva poiskati povezane grafe na n vozliščih s premerom d , ki bodo imeli maksimalno število povezav. Najin cilj je, na podlagi testiranja oz. generiranja, pridobiti kar se da dober vpogled v strukturo teh grafov in posledično ugotoviti, če za njih veljajo kakšne posebne lastnosti. Za majhne vrednosti n in d , se bova problema lotila z generiranjem grafov, za večje pa bova uporabila metodo simulated annealing. Ugotavljala bova tudi učinkovitost najinih metod v odvisnosti od vrednosti n in d .

3. POTEK DELA

Ideja prve faze projekta t.i. exhaustive search-a je, da z generiranjem vseh možnih povezanih grafov na n vozliščih s premerom d , poiščeva tiste, ki imajo maksimalno število povezav. To bova počela za majhne vrednosti n in d . Kako majhne, bo odvisno od časovne zahtevnosti samega algoritma, kajti je pričakovati, da bo že pri ne malo od 5 večjih vrednostih n algoritem počasen. Na podlagi generiranja grafov za različne n in d bova poskušala ugotoviti kakšne lastnosti, tako strukturne kot vizualne, lahko pripiševa tem grafom. Naraščanje/padanje števila povezav v odvisnosti od števila vozlišč oz. premetra bova prikazala tudi s pomočjo grafa, ki se bo morda obnašal podobno kot kakšna znana funkcija, kar bo vsekakor pomagalo pri oceni števila povezav za večje vrednosti n in d . Kot omenjeno bova poskusila najti kakšno formulo za maksimalno število povezav pri številu vozlišč n in premeru d . Tako pridobljene formule, četudi bo morda držala, ne bova dokazovala in jo bova posledično uporabila kot oceno v primeru generiranja grafov. Na koncu bova poleg ugotovitev glede lastnosti grafov v poročilu zapisala tudi pri kako velikih vrednostih n in d je najin algoritem prenehal učinkovito delovati. V drugi fazi projekta se bova problema lotila z metahevrstično metodo simulated annealing. Začela bova z nekim začetnim povezanim grafom G , ki bo ustrezal pogojem n in d , nato pa bova dodala povezavo iz množice povezav komplementa grafa G . V kolikor bo premer grafa $G + e$ ostal isti, imamo nov graf, ki ima isti premer vendar povezavo več. Če bo premer novega grafa manjši od d , pa bova poiskala vozlišči u in v na maksimalni razdalji in odstranjevala povezave iz poti med u in v toliko časa, dokler ne bo premer spet d . Seveda se lahko zgodi, da bo premer večji od d , takrat pa bova spet poiskala vozlišči u in v na maksimalni razdalji in dodajala neke povezave na poti med u in v toliko časa, dokler ne bo premer spet d . Povezave bova morala dodajati med ustreznimi vozlišči. Torej, če bo nov premer $d - 1$, bova dodala povezavo med vozliščema na oddaljenosti 2. Pri tem se zavedava, da z neko verjetnostjo v nekem koraku vzameva graf z manj povezavami, ki pa je morda boljše izhodišče za naprej. Tudi tukaj bova začela na manjših vrednostih, in s tem preveriva, če najin algoritem deluje, nato

pa n in d povečujeva. Tudi v drugi fazi projekta bova pozorna na efektivnost oz. časovno zahtevnost, ter bova ugotovitve glede tega zapisala v poročilu. Algoritme in programe bova v obeh fazah pisala v CoCalc Jupyter notebook-u.

4. KODA

4.1. 1. FAZA - KODA:

```
1      from sage.graphs.graph_generators import graphs
2
3      def najdi_graf_z_premerom(n, d):
4          # Najvecje stevilo povezav in graf z največ povezavami
5          .
6          max_povezave = 0
7          graf_z_max_povezav = None
8
9          # Zanka po vseh povezanih grafih z n vozlišci, ki jih
10             generiramo z uporabo nauty_geng().
11         for G in graphs.nauty_geng(str(n) + "_c"):
12
13             # Premer grafa.
14             premer = G.premer()
15
16             # Ce srečamo graf katerega premer je enak nasemu
17             premeru d
18             if premer == d:
19                 # zabeležimo stevilo povezav
20                 stevilo_povezav = G.size()
21
22                 # Ce je stevilo povezav vecje od trenutnega
23                 maksimuma, posodobi maksimum.
24                 if stevilo_povezav > max_povezave:
25                     max_povezave = stevilo_povezav
26                     graf_z_max_povezav = G.copy()
27
28             return graf_z_max_povezav, max_povezave
29
30     # Primer za neko stevilo vozlišc n in premer d.
31     n = 8
32     d = 3
33
34     # Poiscemo povezan graf z določenim številom vozlišc in
35     premerom, ki bo imel maksimalno stevilo povezav.
36     graf_z_max_povezav, max_povezave = najdi_graf_z_premerom(n
37         , d)
```

```

34 # Ce je graf najden ga prikazemo
35 if graf_z_max_povezav:
36     print(f"Povezan graf z {n} vozlisci in premerom {d} s {
37         max_povezave} povezavami:")
38     print(graf_z_max_povezav)
39     graf_z_max_povezav.show()
40 else:
41     print(f"Graf z {n} vozlisci in premerom {d} ni bil
42         najden.")
43
44 import pandas as pd
45 import matplotlib.pyplot as plt
46
47 rezultati = []
48
49 # Zanka za preiskovanje razlicnih kombinacij n in d, grafe
50   z maksimalnim stevilom povezav shranjujemo v slovar
51 for n in range(1, 10):
52     for d in range(1, n):
53         graf_z_max_povezav, max_povezave =
54             najdi_graf_z_premerom(n, d)
55         rezultat_slovar = {
56             'n': n,
57             'd': d,
58             'max_povezave': max_povezave
59         }
60         rezultati.append(rezultat_slovar)
61
62 # Prikazemo rezultate s tabelo
63 df = pd.DataFrame(rezultati)
64
65 print(df)
66
67 # Prikazemo tudi graf, ki predstavlja maksimalno stevilo
68   povezav v odvisnosti od d za razlicne n
69 plt.figure(figsize=(10, 6))
70 for n in range(1, 9):
71     podskupina = df[df['n'] == n]
72     plt.plot(podskupina['d'], podskupina['max_povezave'],
73             label=f'n={n}')
74 plt.xlabel('premer(d)')
75 plt.ylabel('maksimalno stevilo povezav')
76 plt.legend()
77 plt.title('max_povezave(d) za razlicne n')
78 plt.show()

```

4.2. 2. FAZA - KODA:

```
1     import networkx as nx
2     import matplotlib.pyplot as plt
3     import random
4     from itertools import combinations
5     import math
6
7
8     # Ustvari zaceten graf(drevo) z n vozlicmi in premerom d
9     def zacetni_graf(n, d):
10         if d == 1:
11             G = nx.complete_graph(n)
12             return G
13         else:
14             G = nx.Graph()
15             G.add_nodes_from(range(n))
16             vozlisca = list(G.nodes())
17             for i in range(d): # Poveze d + 1 vozlesc v pot
18                 dolzine d
19                 G.add_edge(vozlisca[i], vozlisca[i + 1])
20             for i in range(d + 1, n): # Nakljucno doda ostale
21                 povezave na notranjih d - 1 vozlesc
22                 izbran = random.randint(1, d - 1)
23                 G.add_edge(vozlisca[i], vozlisca[izbran])
24             return G
25
26     # Presteje stevilo povezav v grafu.
27     def ciljna_funkcija(graf):
28         return len(graf.edges)
29
30     # Najde najkrajso mozno pot v grafu med zacetnim in
31     koncnim vozlescem.
32     def najdi_pot(graf, zacetek, konec):
33         try:
34             pot = nx.shortest_path(graf, source=zacetek,
35                                   target=konec)
36             return pot
37         except nx.NetworkXNoPath:
38             return None
39
40     # Kot argument sprejme graf ter pot iz katere zelimo
41     odstranit povezavo, nato iz nje nakljucno odstrani
42     povezavo.
```

```

40     def odstrani_nakljucno_povezavo_iz_poti_v_grafu(graf, poti)
41         :
42         nakljucni_indeks_povezave = random.randint(1, len(poti)
43             - 1)
44         povezava_za_odstranitev = (poti[
45             nakljucni_indeks_povezave - 1], poti[
46             nakljucni_indeks_povezave])
47         graf.remove_edge(*povezava_za_odstranitev)
48         return graf
49
50     # METAHEVRISTICNI ALGORITEM
51     def simulirano_hlajenje_2_povezavi_razmaka(n, max_iteracij
52         , zacetna_temperatura, stopnja_hlajenja, premer):
53         trenutna_resitev = zacetni_graf(n, premer)
54         najboljsa_resitev = trenutna_resitev.copy()
55         temperatura = zacetna_temperatura
56
57         for iteracija in range(max_iteracij):
58             # Preverimo kaksen je premer, bodisi je vecji ali
59             enak premeru trenutne resitve, bodisi pa je
60             manjsi od 1. V zadnjem primeru je torej
61             nepovezan graf. Dodamo povezavo.
62             if premer <= nx.diameter(trenutna_resitev) or nx.
63                 diameter(trenutna_resitev) < 1:
64                 # Izbere 2 nakljucni vozlišci, ki nista
65                 povezani.
66                 vozlisce1 = random.choice(list(
67                     trenutna_resitev.nodes))
68                 vozlisca_2_razmaka = [vozlisce for vozlisce in
69                     trenutna_resitev.nodes - set([vozlisce1])
70                     if nx.shortest_path_length(
71                         trenutna_resitev, source=vozlisce1, target
72                         =vozlisce) == 2]
73                 vozlisce2 = random.choice(vozlisca_2_razmaka)
74                 # Dodamo povezavo med izbranimi vozlišcema.
75                 nova_resitev = trenutna_resitev.copy()
76                 nova_resitev.add_edge(vozlisce1, vozlisce2)
77                 # Preverimo ali je nov graf tak, da ima vec
78                 povezav. V primeru da je to res,
79                 posodobimo najboljso resitev, sicer pa z
80                 verjetnostjo izberemo ali bomo posodobili
81                 trenutno resitev ali ne.
82                 # Opomba: Lahko pride do izbire "slabsega"
83                 grafa, upamo, da nas bo ta "slabsi" vseeno
84                 pripeljal do boljše resitve v
85                 nadaljevanju.
86                 delta = ciljna_funkcija(nova_resitev) -
87                     ciljna_funkcija(trenutna_resitev)

```

```

66         if (delta > 0 and premer <= nx.diameter(
            nova_resitev)) or random.random() < math.
            exp(-delta / temperatura):
67             trenutna_resitev = nova_resitev.copy()
68             if ciljna_funkcija(nova_resitev) >
                ciljna_funkcija(najboljsa_resitev) and
                premer == nx.diameter(nova_resitev):
69                 najboljsha_resitev = nova_resitev.copy
                    ()
70     else:
71         # Poiscemo kombinacije vozlisc z največjo
            ekscentricnostjo.
72         ekscentricnosti = nx.eccentricity(
            trenutna_resitev)
73         max_ekscentricnost = max(ekscentricnosti.
            values())
74         vozlisca_z_max_ekscentricnostjo = [vozlisce
            for vozlisce, ekscentricnost in
            ekscentricnosti.items() if ekscentricnost
            == max_ekscentricnost]
75         kombinacije_parov = list(combinations(
            vozlisca_z_max_ekscentricnostjo, 2))
76         rezultat = []
77         # Izberemo najbolj oddaljeni vozlisci.
78         for i, j in kombinacije_parov:
79             potencialen_rezultat = najdi_pot(
                trenutna_resitev, i, j)
80             if potencialen_rezultat and len(
                potencialen_rezultat) > len(rezultat):
81                 rezultat = potencialen_rezultat
82         nova_resitev = trenutna_resitev.copy()
83         nova_resitev =
            odstrani_nakljucno_povezavo_iz_poti_v_grafu
            (trenutna_resitev.copy(), rezultat)
84         # Preverimo ali je nov graf tak, da ima vec
            povezav. V primeru da je to res,
            posodobimo najboljso resitev, sicer pa z
            verjetnostjo izberemo ali bomo posodobili
            trenutno resitev ali ne.
85         # Opomba: Lahko pride do izbire "slabsega"
            grafa, upamo, da nas bo ta "slabsi" vseeno
            pripeljal do boljse resitve v
            nadaljevanju.
86         delta = ciljna_funkcija(nova_resitev) -
            ciljna_funkcija(trenutna_resitev)
87         # IF pogoj je vedno izpolnjen, ko je novi graf
            povezan, prvi del pogoja pustimo zgolj za
            voljo testiranja.

```

```

88         if ((delta > 0 and premer <= nx.diameter(
            nova_resitev)) or random.random() < math.
            exp(-delta / temperatura)) and nx.
            is_connected(nova_resitev):
89             trenutna_resitev = nova_resitev.copy()
90             if ciljna_funkcija(nova_resitev) >
                ciljna_funkcija(najboljsa_resitev) and
                premer == nx.diameter(nova_resitev):
91                 najboljsa_resitev = nova_resitev.copy()
                    ()
92             # Znizamo(ohladimo) temperaturo po stopnji
                hlajenja.
93             temperatura *= stopnja_hlajenja
94
95         return najboljsa_resitev
96
97     # Prikaz delovanja algoritma na primeru.
98     st_vozlisc = 30
99     max_iteracij = 1000
100     zacetna_temperatura = 1.0
101     stopnja_hlajenja = 0.95
102     zeljen_premer = 7
103
104     najboljsi_graf = simulirano_hlajenje_2_povezavi_razmaka(
        st_vozlisc, max_iteracij, zacetna_temperatura,
        stopnja_hlajenja, zeljen_premer)
105     print(f"Stevilo povezav v najboljšem generiranem grafu: {
        ciljna_funkcija(najboljsi_graf)}")
106
107     # Graf se prikazemo.
108     plt.figure(figsize=(8, 8))
109     nx.draw(najboljsi_graf, with_labels=True, font_weight='
        bold', node_color='skyblue', node_size=800, font_size
        =10)
110     plt.show()

```


5. UGOTOVITVE

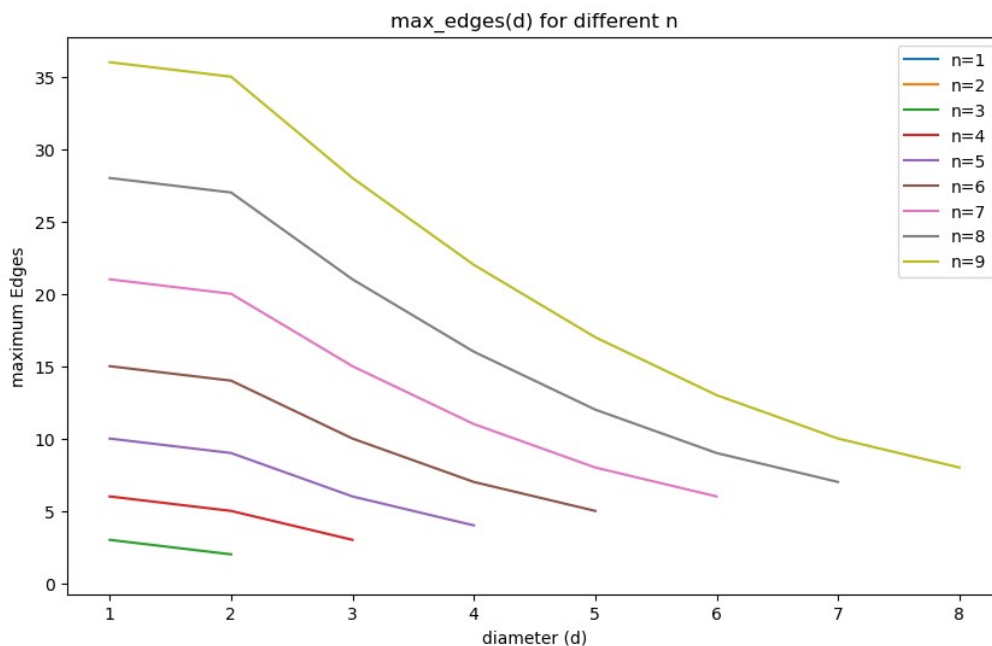
5.1. 1. FAZA - UGOTOVITVE:

Za $d = 1$ ugotovimo, da je ne glede na izbiro števila vozlišč n , iskani graf ravno polni graf in ima posledično $\frac{n(n-1)}{2}$ povezav. V naslednjem koraku hitro ugotovimo, da se pri $d = 2$ število povezav zmanjša le za 1, saj se z odstranitvijo katere koli poljubne povezave v polnem grafu premer poveča na $d = 2$ in ker smo za to potrebovali odstraniti le eno samo povezavo je največje možno število povezav v grafu z n točkami in premerom $d = 2$ enako $\frac{n(n-1)}{2} - 1$. Podobno opazimo, da so grafi za premere $d = n - 1$ ravno drevesa s stopnjo 2 in je zato število povezav enako $n - 1$. Tako nas pri dani nalogi v resnici zanimajo predvsem grafi z $d \in \{3, \dots, n - 2\}$. V prvi fazi sva pričela reševati z opazovanjem in računanjem grafov z manjšim številom vozlišč n , pri tem sva si pomagala tudi s kodo iz 4.1.

Napisani algoritem je z uporabo funkcije nauty geng generiral vse povezane grafe na n vozliščih, izločeval tiste, katerih premer ni bil enak d , ter posodabljal spremenljivko z maksimalnim številom povezav ter tem povezavam ustreznemu grafu. Podatke o največjem možnem številu povezav za grafe do 10 vozlišč sva zbrala v tabeli 1 in vrednosti prikazala na spodnjem grafu.

n\d	Število povezav								
	1	2	3	4	5	6	7	8	9
2	1								
3	3	2							
4	6	5	3						
5	10	9	6	4					
6	15	14	10	7	5				
7	21	20	15	11	8	6			
8	28	27	21	16	12	9	7		
9	36	35	28	22	17	13	10	8	
10	45	44	36	29	23	18	14	11	9

TABELA 1. Število povezav glede na število vozlišč in premer.

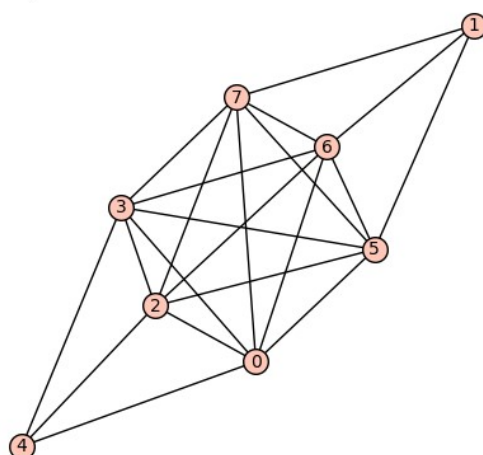


SLIKA 1. Maksimalno število povezav v odvisnosti od d pri različnih n .

Z opazovanjem tabele sva na podlagi vzorca uspela za grafe z $d > 1$ zapisati formulo, ki nama pove maksimalno število povezav v grafu z n vozlišči in premerom d : $\frac{(n-d+1)(n-d)}{2} + n - 2$. Te formule ne bova dokazovala in jo bova v nadaljnjem raziskovanju uporabljala kot oceno, saj vanjo brez dokaza ne moreva biti popolnoma prepričana.

Z nadaljnjim opazovanjem generiranih grafov sva opazila, da vsi grafi vsebujejo poln podgraf velikosti $n - d + 1$. Na grafu prikazanem spodaj se to lepo vidi.

Out[2]: Connected graph with 8 vertices and diameter 3 with 21 edges:
Graph on 8 vertices



SLIKA 2. Graf z 8 vozlišči in premerom 3, ki vsebuje poln podgraf velikosti 6.

Glede učinkovitosti sva ugotovila, da je najin algoritem učinkovit za grafe z številom vozlišč do 9. Od tod naprej traja enostavno preveč časa. Že pri številu vozlišč enako 8, je povezanih grafov kar 251548592

5.2. 2. FAZA - TESTIRANJE:

n/d	Število povezav																																												
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	
2	1																																												
3	3	2																																											
4	6	5	3																																										
5	10	9	6	4																																									
6	15	14	10	7	5																																								
7	21	20	15	11	8	6																																							
8	28	27	21	16	12	9	7																																						
9	36	35	28	22	17	13	10	8																																					
10	45	44	36	29	23	18	14	11	9																																				
11	55	54	45	37	30	23	19	15	12	10																																			
12	66	65	55	46	38	30	22	20	16	13	11																																		
13	78	77	66	56	47	31	30	23	20	17	14	12																																	
14	91	90	78	67	53	48	39	29	27	20	17	15	13																																
15	105	104	91	79	65	46	49	41	27	24	21	19	16	14																															
16	120	119	105	92	76	63	47	38	32	35	29	24	20	17	15																														
17	136	135	120	103	93	78	52	55	45	34	32	27	23	21	18	16																													
18	153	152	136	121	107	63	58	61	45	42	39	31	26	24	22	19	17																												
19	171	170	153	133	116	90	63	62	57	43	39	37	31	28	27	22	20	18																											
20	190	189	171	154	138	107	82	68	61	56	51	40	42	31	28	26	25	21	19																										
21	210	209	190	172	155	133	99	75	62	69	50	57	45	39	32	31	27	24	22	20																									
22	231	230	210	179	146	144	104	87	75	64	77	57	49	39	35	37	30	28	25	23	21																								
23	253	252	231	203	192	169	157	99	72	74	68	62	54	49	43	41	36	31	28	26	24	22																							
24	276	275	253	226	207	123	147	105	100	91	69	69	57	56	48	41	41	36	37	30	27	25	23																						
25	300	299	276	248	219	213	148	112	111	100	73	79	58	62	54	50	44	39	37	33	30	28	26	24																					
26	325	324	300	282	237	146	156	174	140	91	92	90	77	71	69	51	46	48	42	37	34	32	29	27	25																				
27	351	350	325	294	210	172	173	135	124	113	123	88	93	67	67	54	60	47	44	44	39	35	33	30	28	26																			
28	378	377	351	314	223	222	151	190	180	127	134	88	79	79	76	66	63	58	50	43	45	39	36	33	31	29	27																		
29	406	405	378	343	281	194	168	200	159	115	137	102	81	87	71	74	65	57	58	51	47	42	40	39	34	32	30	28																	
30	435	434	406	364	294	321	186	163	152	145	122	109	96	99	89	82	71	59	56	53	60	49	42	40	38	36	33	31	29																
31	465	464	435	392	286	241	248	175	147	178	124	109	104	90	103	89	90	75	65	56	51	53	47	43	41	41	37	34	32	30															
32	496	495	465	412	363	281	231	184	178	191	209	133	135	118	100	85	85	76	74	66	60	58	54	47	47	41	42	37	35	33	31														
33	528	527	496	450	390	256	229	202	208	211	172	134	132	107	100	97	86	100	78	66	66	63	57	60	47	46	43	41	38	36	34	32													
34	561	560	528	489	370	309	261	224	187	205	145	204	120	117	117	115	92	92	87	102	72	67	69	54	52	51	49	46	42	40	37	35	33												
35	595	594	561	480	399	328	293	219	240	187	158	138	138	139	123	107	97	91	79	78	83	72	65	62	60	54	52	48	45	43	41	38	36	34											
36	630	629	595	547	394	295	281	239	282	192	157	158	163	135	136	114	108	98	89	78	79	76	69	69	60	60	55	51	49	46	44	41	40	37	35										
37	666	665	630	579	390	433	288	245	236	195	203	196	147	138	133	123	121	100	108	96	80	82	74	77	78	62	63	54	51	50	49	45	43	40	38	36									
38	703	702	666	599	514	363	326	306	219	215	197	160	151	164	133	132	109	111	119	99	90	88	81	83	71	64	65	64	59	53	50	50	45	43	41	39	37								
39	741	740	703	650	530	360	330	277	259	218	219	257	178	159	156	133	116	118	106	108	108	104	90	81	74	79	77	64	60	59	54	51	48	47	44	42	40	38							
40	780	779	741	656	493	480	318	304	256	283	204	200	176	224	176	145	123	147	117	117	102	103	89	88	85	82	90	67	62	64	58	63	57	49	48	46	43	41	39						
41	820	819	780	695	564	489	356	316	283	342	248	237	212	162	178	147	177	122	135	126	133	108	99	91	99	81	95	71	71	67	64	61	55	54	51	49	46	45	42	40					
42	861	860	816	722	561	459	370	351	271	296	225	221	194	184	186	165	138	153	148	134	111	103	103	100	101	90	85	80	72	74	67	68	60	57	54	51	50	47	45	43	41				
43	903	902	852	737	601	477	390	361	296	249	267	243	206	237	210	176	151	172	138	142	126	116	114	105	91	97	91	83	82	86	71	69	63	60	58	54	53	52	48	46	44	42	43		
44	946	945	886	770	569	466	419	379	309	297	246	265	196	221	178	196	168	162	146	141	138	119	127	102	105	98	96	99	86	74	75	69	70	64	61	57	56	54	51	49	47	45	43		
45	990	989	924	811	639	463	495	382	333	307	279	257	225	237	202	234	138	165	169	150	132	131	124	116	109	102	96	85	88	82	78	76	72	71	66	61	59	57	55	53	50	48	46	44	

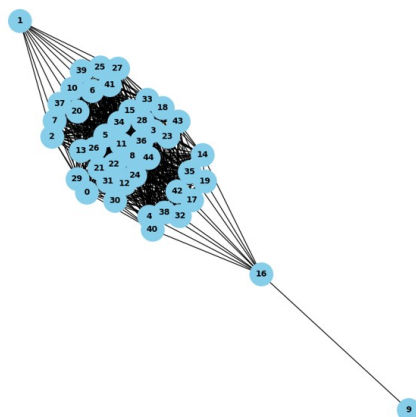
n\ d	Število povezav																																													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44		
2	1																																													
3	3	2																																												
4	6	5	3																																											
5	10	9	6	4																																										
6	15	14	10	7	5																																									
7	21	20	15	11	8	6																																								
8	28	27	21	16	12	9	7																																							
9	36	35	28	22	17	13	10	8																																						
10	45	44	36	28	23	18	14	11	9																																					
11	55	54	45	37	30	21	18	15	12	10																																				
12	66	65	55	46	38	31	25	18	16	13	11																																			
13	78	77	66	56	47	31	30	23	21	17	14	12																																		
14	91	90	78	67	54	46	32	27	24	22	18	15	13																																	
15	105	104	91	79	68	51	39	33	34	28	23	19	16	14																																
16	120	119	105	88	80	69	44	48	37	31	25	22	20	17	15																															
17	136	135	120	100	93	77	68	60	51	35	30	27	23	21	18	16																														
18	153	152	136	121	99	88	80	57	45	42	34	29	31	24	21	19	17																													
19	171	170	153	137	119	106	65	62	52	49	42	44	32	27	25	22	20	18																												
20	190	189	171	154	132	122	83	70	80	53	48	44	41	33	28	26	24	21	19																											
21	210	209	190	168	145	127	88	78	69	54	56	49	43	36	32	28	27	25	22	20																										
22	231	230	210	179	148	130	116	113	72	68	69	58	51	42	35	36	32	27	26	23	21																									
23	253	252	231	193	136	174	107	94	92	73	71	58	60	47	42	38	34	33	29	26	24	22																								
24	276	275	253	232	188	151	155	122	96	75	81	63	52	50	47	47	38	34	32	30	27	25	23																							
25	300	299	276	244	221	132	134	124	121	94	77	79	70	66	64	49	41	42	38	34	30	28	26	24																						
26	325	324	300	263	196	176	149	121	106	130	89	85	104	63	66	54	53	43	38	38	34	31	29	27	25																					
27	351	350	325	289	258	244	147	126	112	110	92	94	74	64	69	62	51	53	43	41	38	40	32	31	28	26																				
28	378	377	351	306	290	172	188	148	147	124	102	103	76	77	66	59	55	51	59	44	41	39	36	36	31	29	27																			
29	406	405	378	334	275	263	165	186	147	128	102	105	92	77	78	71	64	57	61	49	45	41	41	37	35	32	30	28																		
30	435	434	406	379	329	193	185	168	166	129	131	121	102	92	97	79	70	60	55	54	56	50	44	41	37	35	33	31	29																	
31	465	464	435	403	380	252	227	200	143	182	133	126	102	108	101	80	96	76	70	60	58	55	51	44	42	41	37	35	32	30																
32	496	495	465	417	345	238	207	197	155	161	157	124	112	115	104	95	83	84	66	70	61	60	53	50	46	44	44	38	35	33	31															
33	528	527	496	448	397	304	253	203	172	171	176	152	120	116	94	99	90	98	90	85	70	60	60	58	49	47	43	41	41	36	34	32														
34	561	560	528	478	434	325	313	238	194	198	179	138	149	123	113	103	93	89	84	73	70	74	61	56	51	50	49	43	42	39	37	35	33													
35	595	594	561	478	429	352	310	211	182	242	161	155	159	132	112	136	105	97	80	77	88	73	75	62	67	62	61	49	45	43	43	38	36	34												
36	630	629	595	544	576	469	294	212	218	220	140	149	145	122	153	107	93	99	86	83	79	70	69	61	66	60	50	51	46	43	41	39	37	35												
37	666	665	630	571	478	355	310	249	205	198	188	167	167	145	127	120	115	109	111	94	88	84	81	69	72	63	61	54	52	50	46	45	42	40	38	36										
38	703	702	666	614	517	417	287	249	212	209	184	215	160	144	141	120	112	116	101	94	95	87	86	83	77	71	72	50	55	57	51	49	46	43	41	39	37									
39	741	740	703	647	543	368	340	261	256	233	223	191	179	164	154	136	123	135	114	109	107	94	89	82	85	68	67	64	59	59	56	52	48	46	44	42	40	38								
40	780	779	741	695	551	382	466	292	266	255	209	207	195	190	158	143	163	120	113	114	105	106	91	86	83	78	72	65	70	61	56	57	54	54	47	45	43	41	39							
41	820	819	780	709	588	379	384	293	277	331	249	238	208	178	201	137	142	132	125	116	113	116	112	103	81	81	75	73	67	62	67	59	57	55	50	48	46	44	42	40						
42	861	860	819	720	571	443	427	384	283	293	243	219	212	180	179	182	137	139	132	118	116	115	108	101	85	84	88	85	73	72	68	64	62	57	57	50	47	45	43	41						
43	903	902	857	755	713	509	426	337	301	263	236	284	221	185	186	155	146	157	132	130	124	128	128	98	103	91	95	79	77	72	70	67	68	62	57	56	53	50	49	47	44	42				
44	946	945	893	781	608	477	409	333	319	269	254	285	197	205	179	197	176	169	153	129	130	114	114	108	105	102	87	84	82	85	73	72	71	67	65	60	57	52	49	47	45	43				
45	990	989	916	827	682	501	450	361	384	303	260	258	256	222	222	180	171	174	155	154	138	139	121	123	111	103	94	96	91	83	79	79	68	68	63	62										

n\ d	Števílo povezav																																													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44		
2	1																																													
3	3	2																																												
4	6	5	3																																											
5	10	9	6	4																																										
6	15	14	10	7	5																																									
7	21	20	15	11	8	6																																								
8	28	27	21	16	12	9	7																																							
9	36	35	28	22	17	13	10	8																																						
10	45	44	36	29	23	18	14	11	9																																					
11	55	54	45	37	30	24	17	15	12	10																																				
12	66	65	55	46	38	26	22	18	16	13	11																																			
13	78	77	66	56	47	39	31	26	21	17	14	12																																		
14	91	90	78	67	57	48	32	28	23	22	18	15	13																																	
15	105	104	91	79	68	56	37	33	30	24	21	19	16	14																																
16	120	119	105	92	80	61	52	40	42	31	24	22	20	17	15																															
17	136	135	120	106	93	57	52	44	39	35	28	26	23	20	18	16																														
18	153	152	136	113	103	90	56	51	54	42	34	29	28	24	21	19	17																													
19	171	170	153	129	118	81	85	74	49	48	40	33	31	29	25	23	20	18																												
20	190	189	171	148	117	111	89	80	60	65	44	44	46	33	28	25	23	21	19																											
21	210	209	190	166	155	135	82	75	67	63	53	47	45	40	33	31	26	25	22	20																										
22	231	230	210	187	157	104	94	85	90	67	62	51	45	42	46	35	32	27	26	23	21																									
23	253	252	231	205	180	159	101	108	101	67	71	60	50	49	40	35	30	29	26	24	22																									
24	276	275	253	228	195	127	130	122	91	95	74	69	54	50	47	42	39	34	31	30	27	25	23																							
25	300	299	276	254	208	177	134	133	108	104	73	67	67	66	53	49	47	40	38	33	31	28	26	24																						
26	325	324	300	277	236	190	214	125	108	104	96	84	73	62	59	48	48	44	39	37	36	31	30	27	25																					
27	351	350	325	301	248	172	181	155	121	120	99	78	75	73	68	65	56	49	43	42	40	35	33	30	28	26																				
28	378	377	351	306	289	184	192	166	131	101	97	98	102	78	69	62	56	52	49	48	43	37	37	34	31	29	27																			
29	406	405	378	344	288	218	176	153	148	121	101	108	103	82	85	69	75	66	61	51	47	43	44	38	35	33	30	28																		
30	435	434	406	339	344	246	194	175	230	120	121	94	96	108	89	84	71	67	55	58	52	47	46	40	39	36	33	31	29																	
31	465	464	435	383	308	216	184	174	142	178	133	110	117	104	88	78	85	71	57	61	53	67	46	46	40	39	39	34	32	30																
32	496	495	465	430	376	266	199	185	170	155	148	132	129	113	106	89	103	94	66	60	62	60	53	48	46	42	40	38	35	33	31															
33	528	527	496	406	357	346	214	195	173	174	176	149	154	99	118	89	85	96	75	76	73	65	57	51	49	45	43	41	39	36	34	32														
34	561	560	528	482	406	269	229	219	200	165	171	171	129	113	105	106	92	83	90	73	73	75	59	54	56	51	47	46	42	40	37	35	33													
35	595	594	561	509	341	293	323	200	195	185	130	198	143	167	115	105	93	96	84	90	79	71	65	64	57	57	54	48	45	42	40	38	36	34												
36	630	629	595	531	470	349	332	227	199	183	143	167	126	128	130	108	101	99	85	79	76	72	66	61	60	56	52	48	45	43	41	40	37	35												
37	666	665	630	562	520	436	271	267	236	232	199	156	159	135	133	112	123	92	97	94	78	74	73	65	63	62	56	52	53	46	44	42	40	38	36											
38	703	702	666	610	487	337	294	273	274	196	223	175	186	157	132	127	109	127	96	101	108	93	82	78	79	66	65	60	61	55	50	48	45	46	41	39	37									
39	741	740	703	625	558	376	334	302	291	233	217	206	194	154	139	138	162	121	115	109	102	95	82	82	82	75	73	64	60	58	55	52	51	48	45	43	40	38								
40	780	779	741	686	539	484	343	303	333	244	317	201	216	165	156	157	137	134	122	108	104	99	93	87	84	88	75	68	65	60	59	57	51	52	48	46	43	41	39							
41	820	819	780	694	548	514	368	355	309	265	279	201	187	189	191	168	160	126	133	138	112	107	98	95	85	88	84	80	76	69	69	63	61	56	56	50	50	46	44	42	40					
42	861	860	819	728	614	415	385	351	309	247	258	207	205	192	173	171	164	163	127	138	112	105	100	99	95	107	94	85	78	70	70	68	64	60	53	52	50	47	45	43	41					
43	903	902	855	773	664	412	431	354	297	339	285	204	197	193	193	197	142	153	124	136	119	112	106	97	115	86	78	78	72	71	75	69	64	59	55	56	51	48	46	44	42					
44	946	945	891	801	617	556	425	344	353	313	240	234	257	209	176	160	195	157	145	145	133	125	118	100	117	100	102	94	84	83	74	74	73	68	66	61	58	55	54	52	50	47	45	43		
45	990	989	919	806	635	552	461	390	383	276	294	260	242	239	190	203																														

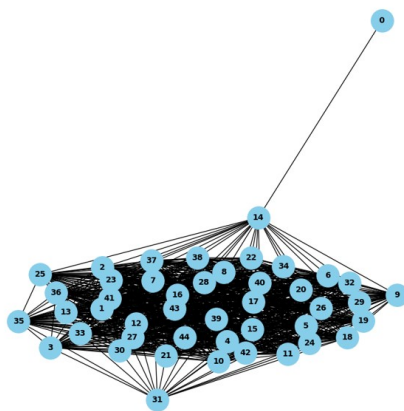
n\ d	Število povezav																																													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44		
2	1																																													
3	3	2																																												
4	6	5	3																																											
5	10	9	6	4																																										
6	15	14	10	7	5																																									
7	21	20	15	11	8	6																																								
8	28	27	21	16	12	9	7																																							
9	36	35	28	22	17	13	10	8																																						
10	45	44	36	29	23	18	14	11	9																																					
11	55	54	45	37	30	24	19	15	12	10																																				
12	66	65	55	46	38	31	25	20	16	13	11																																			
13	78	77	66	56	47	39	32	26	21	17	14	12																																		
14	91	90	78	67	57	48	40	33	27	22	18	15	13																																	
15	105	104	91	79	68	58	49	41	34	28	23	19	16	14																																
16	120	119	105	92	80	69	59	50	42	35	29	24	20	17	15																															
17	136	135	120	106	93	81	70	60	51	43	36	30	25	21	18	16																														
18	153	152	136	121	107	94	82	71	61	52	44	37	31	26	22	19	17																													
19	171	170	153	137	122	108	95	83	72	62	53	45	38	32	27	23	20	18																												
20	190	189	171	154	138	123	109	96	84	73	63	54	46	39	33	28	24	21	19																											
21	210	209	190	172	155	139	124	110	97	85	74	64	55	47	40	34	29	25	22	20																										
22	231	230	210	191	173	156	140	125	111	98	86	75	65	56	48	41	35	30	26	23	21	22																								
23	253	252	231	211	192	174	157	141	126	112	99	87	76	66	57	49	42	36	31	27	24	22	23																							
24	276	275	253	232	212	193	175	158	142	127	113	100	88	77	67	58	50	43	37	32	28	25	23	24																						
25	300	299	276	254	233	213	194	176	159	143	128	114	101	89	78	68	59	51	44	38	33	29	26	24	25																					
26	325	324	300	277	255	234	214	195	177	160	144	129	115	102	90	79	69	60	52	45	39	34	30	27	25	26																				
27	351	350	325	301	278	256	235	215	196	178	161	145	130	116	103	91	80	70	61	53	46	40	35	31	28	26	27																			
28	378	377	351	326	302	279	257	236	216	197	179	162	146	131	117	104	92	81	71	62	54	47	41	36	32	29	27	28																		
29	406	405	378	352	327	303	280	258	237	217	198	180	163	147	132	118	105	93	82	72	63	55	48	42	37	33	30	28	29																	
30	435	434	406	379	353	328	304	281	259	238	218	199	181	164	148	133	119	106	94	83	73	64	56	49	43	38	34	31	29	30																
31	465	464	435	407	380	354	329	305	282	260	239	219	200	182	165	149	134	120	107	95	84	74	65	57	50	44	39	35	32	30																
32	496	495	465	436	408	381	355	330	306	283	261	240	220	201	183	166	150	135	121	108	96	85	75	66	58	51	45	40	36	33	31															
33	528	527	496	466	437	409	382	356	331	307	284	262	241	221	202	184	167	151	136	122	109	97	86	76	67	59	52	46	41	37	34	32														
34	561	560	528	497	467	438	410	383	357	332	308	285	263	242	222	203	185	168	152	137	123	110	98	87	77	68	60	53	47	42	38	35	33													
35	595	594	561	529	498	468	439	411	384	358	333	309	286	264	243	223	204	186	169	153	138	124	111	99	88	78	69	61	54	48	43	39	36	34												
36	630	629	595	562	530	499	469	440	412	385	359	334	310	287	265	244	224	205	187	170	154	139	125	112	100	89	79	70	62	55	49	44	40	37	35											
37	666	665	630	596	563	531	500	470	441	413	386	360	335	311	288	266	245	225	206	188	171	155	140	126	113	101	90	80	71	63	56	50	45	41	38	36										
38	703	702	666	631	597	564	532	501	471	442	414	387	361	336	312	289	267	246	226	207	189	172	156	141	127	114	102	91	81	72	64	57	51	46	42	39	37									
39	741	740	703	667	632	598	565	533	502	472	443	415	388	362	337	313	290	268	247	227	208	190	173	157	142	128	115	103	92	82	73	65	58	52	47	43	40	38								
40	780	779	741	704	668	633	599	566	534	503	473	444	416	389	363	338	314	291	269	248	228	209	191	174	158	143	129	116	104	93	83	74	66	59	53	48	44	41	39							
41	820	819	780	742	705	669	634	600	567	535	504	474	445	417	390	364	339	315	292	270	249	229	210	192	175	159	144	130	117	105	94	84	75	67	60	54	49	45	42	40						
42	861	860	820	781	745	706	670	635	601	568	536	505	475	446	418	391	365	340	316	293	271	250	230	211	193	176	160	145	131	118	106	95	85	76	68	61	55	50	46	43	41					
43	903	902	861	821	782	746	707	671	636	602	569	537	506	476	447	419	392	366	341	317	294	272	251	231	212	194	177	161	146	132	119	107	96	86	77	69	62	56	51	47	44	42				
44	946	945	903	862	822	783	745	708	672	637	603	570	538	507	477	448	420	393	367	342	318	295	273	252	232	213	195	178	162	147	133	120	108	97	87	78	70	63	57	52	48	45	43			
45	990	9																																												

[illegible][illegible]

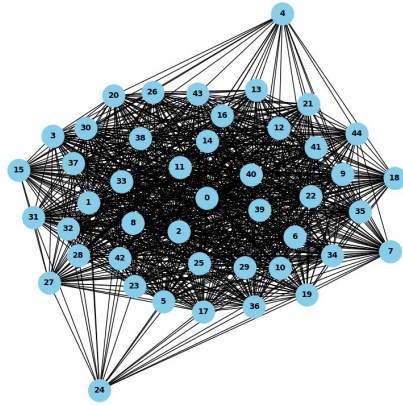
Gornje tabele so rezultat testiranja pri različnih parametrih (navedenih v opisu tabel), spodaj pa so slike nekaterih grafov iz tabel. Potrdiva lahko hipotezo o vsebovanosti kompletnih podgrafov ter vidiva, da najina ocena za maksimalno število povezav drži. Pri 1000 iteracijah je metahevrstičen algoritem prenehal biti učinkovit, a sva vseeno naredila še testiranje pri 2000 iteracijah, ter dobila enako ustrezne rezultate.



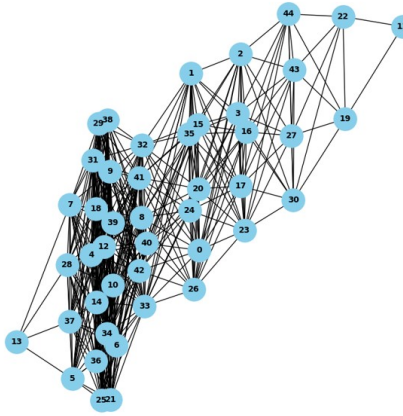
SLIKA 3. $n = 45$, $d = 5$, št. povezav = 706 in 1000 iteracij, št. povezav po formuli = 863.



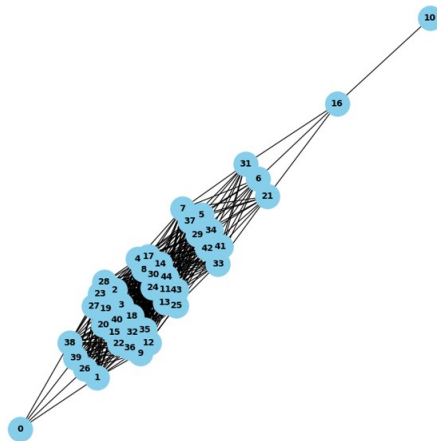
SLIKA 4. $n = 45$, $d = 4$, št. povezav = 859 in 1000 iteracij, št. povezav po formuli = 904.



SLIKA 5. $n = 45$, $d = 3$, št. povezav = 914 in 1000 iteracij, št. povezav po formuli = 946.



SLIKA 6. $n = 45$, $d = 8$, št. povezav = 425 in 2000 iteracij, št. povezav po formuli = 746.



SLIKA 7. $n = 45$, $d = 7$, št. povezav = 571 in 2000 iteracij, št. povezav po formuli = 784.