UNIVERZA V LJUBLJANI
FAKULTETA ZA MATEMATIKO IN FIZIKO

Finančna matematika – 1. stopnja

Peter Milivojević

IGRE USTVARJANJA OMREŽIJ

Delo diplomskega seminarja

Mentor: prof. dr. Sergio Cabello Justo

Ljubljana, 2024

Kazalo

Igre ustvarjanja omrežij

Povzetek

...

Network creation games

Abstract

...

Math. Subj. Class. (2020): ..., ...
Ključne besede: ..., ...
Keywords: ..., ...

1 Uvod

Namen diplomske naloge se je spoznati z igrami ustvarjanja omrežja s poudarkom na dve osnovni verziji tega problema. V igri ustvarjanja omrežja imamo igralce, predstavljene kot vozlišča v grafu, ki želijo s 'sebično' izbiro svoje strategije izboljšati svoj položaj. Običajno ima vsak igralec dva sebična cilja. Prvi cilj je minimizirat stroške ustvarjanja povezav (omrežja) in drugi je minimizirat razdaljo, do ostalih vozlišč (strošek uporabe omrežja).

V osnovnih igrah omrežij predpostavimo, da se ne da primerjati cene ustvarjanja in vzdrževanja povezav. Zato se omejimo na že v naprej podane grafe (omrežja), kjer lahko vozlišča (igralci) le zamenjajo svoje povezave ali v posebnem primeru odstranijo povezavo, tako da zamenjajo pvezavo za že obstoječo povezavo, s čimer se ena povezava izbriše, saj se bomo ukvarjali le zgrafi brez zank in dvojnih povezav. Ne morejo pa ustvarit novih povezav.

2 Teoretična podlaga

Da bomo lahko razumeli obnašanje igralcev in lastnosti nastalih omrežjih bomo prvo obnovili/postavili teoretične temelje. Ukvarjali se bomo izključno z povezanimi enostavnimi grafi.

Za lažje razumevanje nadaljnih izrekov, lem, trditev, posledic in rezultatov bomo ponovili nekaj definicij o grafih.

Definicija 2.1. $Graf$ je urejen par G = (V, E), kjer je V neprazna množica točk grafa G in E množica povezav grafa G, pričemer je vsaka povezava par točk.

Definicija 2.2. $Graf$ G je povezan, če za vsak par voclišč $u, v \in V(G)$ obstaja pot od u do v.

Definicija 2.3. Naj bo G povezan graf in $v \in V(G)$. Stopnja točke v je enaka vsoti števila povezav, ki imajo to točko za krajišče (in dvojnega števila zank v tej točki). Označimo jo z $deg(v)$.

Definicija 2.4. Naj bo G povezan graf in $u, v \in V(G)$. Razdalja $d(u, v)$ je dolžina najkrajše poti med vozliščema u in v (t.j. razdalja med u in v) v grafu G.

Definicija 2.5. Naj bo G povezan graf. Premer grafa G je definiran kot $\text{diam}(G) = \max_{u,v \in V(G)} d(u, v)$, kjer je $d(u, v)$ razdalja med vozliščema u in v v grafu G.

Definicija 2.6. Naj bo G povezan graf. Lokalni premer točke v grafa G je definiran kot $\text{diam}(G) = \max_{u \in V(G)} d(u, v)$, kjer je $d(u, v)$ razdalja med vozliščema u in v v grafu G.

Definicija 2.7. Povezan graf G ima prerezno vozlišče v, če graf $G - v$ ni povezan.

Weinerjev indeks nam bo pomagal z kasnejšimi dokazi.

Definicija 2.8. Naj bo G povezan graf z n vozlišči. Weinerjev indeks $W = W(G)$ je definiran kot vsota vseh razdalj med vozlišči.

$$W(G) = \sum_{i=1}^{n} \sum_{j=1}^{i} d_{ij} = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij}$$

kjer d_{ij} označuje dolžino najkrajše poti med vozliščem i in j.

3 Igra

Pomembno za razumevanje kasnejših delov diplomske naloge je tudi razumevanje kaj je igra v smislu teorije iger.

!!!!!!!!!

Definicija 3.1. Strateška igra s funkcijo preferenc je trojica $(N, (A_i)_{i \in N}, (u_i)_{i \in N})$ pri čemer:

- N je množica igralcev, v našem primeru je to število točk v grafu
- Za vsakega igralca $i \in N$ je A_i neprazna množica njegovih akcij, med katerimi v danem trenutku izbera igralec i
- Za vsakega igralca $i \in N$ je u_i funkcija preferenc na A_i. Torej $u_i : A_i \to \mathbb{R}$ v splošnem in $u_i : A_i \to \mathbb{N}$ v našem primeru.

Za funkcijo preferenc lahko zapišemo sledeče relacije.
$\forall a, b \in A_i :$

- $(u(a) \geq u(b) \Rightarrow a)$ je vsaj tako dobro kot b
- $(u(a) > u(b) \Rightarrow a)$ je boljše kot b
- $(u(a) = u(b) \Rightarrow b)$ indiferenten med a in b

4 Ravnovesje

V tem delu bomo definirali pravila igre in pogoje za nastanek ravnovesji.

Definicija 4.1. Graf je v *ravnotežju glede na vsoto razdalj*, če za vsako povezavo vw in za vsako vozlišče w' zamenjava povezave vw z povezavo vw' ne zmanjša celotne vsote razdalj od vozlišča v do vseh ostalih vozlišč.

Definicija 4.2. Graf je v *ravnotežju glede na maksimalno razdaljo*, če za vsako povezavo vw in za vsako vozlišče w' zamenjava povezave vw z povezavo vw' ne zmanjša lokalnega premera vozlišča v. Nadalje odstranitev povezave vw poveča lokalni premer vozlišča v.

Definicija 4.3. Naj bo G povezan graf. Graf G je *kritičen za odstranitev povezave*, če odstranitev katere koli povezave $uv \in E(G)$ poveča lokalni premer vozlišča v in vozlišča u.

Definicija 4.4. Naj bo G povezan graf. Graf G je *stabilen za dodajanje povezave*, če dodajanje katere koli povezave $uv \in E(G)$ ne zmanjša lokalnega premera vozlišča v in vozlišča u.

5 Cena anarhije in cena stabilnosti

V delu se bomo ukvarjalise tudi z ceno anarhije (PoA) in ceno stabilnosti (PoS). Ceni anarhije in stabilnosti merita, kako se učinkovitost sistema poslabša zaradi sebičnega vedenja njegovih agentov. Cena anarhije je razmerje med vrednostjo/ceno, iz socialnega vidika (skupnega vidika vseh igralcev), najslabšega ravnovesja in socialno ceno optimalnega izida Cena stabilnosti igre pa je razmerje med najboljšo socialno ceno ravnovesja in socialno ceno optimalnega izida.
Socialno ceno grafa bomo definirali kot vsoto cen vseh vozlišč. Za igro vsote bomo tako socialno ceno definirali kot $SC(G) = 2W(G)$, za igro lokalnega premera pa kot $SC(G) = \sum_{i=1}^{n} \epsilon_G(i)$. Tako lahko definiramo sledeče:

Definicija 5.1. $PoA = \frac{\text{Socialna cena najslabšega ravnovesja}}{\text{Socialna cena optimalne postavitve}} = \frac{\max_{G \in R} SC(G)}{\min_{G \in A} SC(G)}$

Definicija 5.2. $PoS = \frac{\text{Socialna cena najboljšega ravnovesja}}{\text{Socialna cena optimalne postavitve}} = \frac{\min_{G \in R} SC(G)}{\min_{G \in A} SC(G)}$

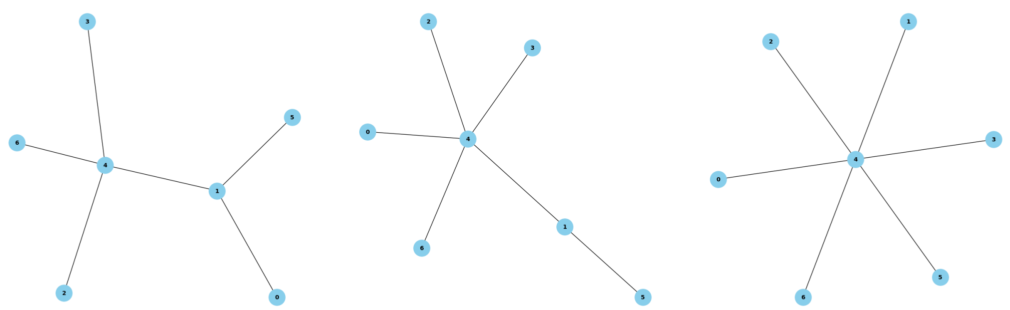
Kjer A predstavlja množico vseh možnih povezanih grafov z $|V(G)| = n$ vozlišči in manj ali enako kot $|E(G)| = m$ povezavami. Množica $R \subseteq A$ pa predstavlja množico vseh ravnovesnih grafov, ki lahko nastanejo iz začetnih grafov z n vozlišči in m povezavami.

6 Teoretični del diplomske naloge

Nekoliko zahtevnejši in bolj raznoliki grafi od polnih grafov so drevesa.

Izrek 6.1. *Če je ravnovesni graf za skupno vsoto razdalj v preprosti igri ustvarjanja omrežja drevo, potem ima premer največ* 2 *in je kot tak zvezda.*

Dokaz. Dokaza se bomo lotili z protislovjem. Predpostavimo, da je ravnovesni graf drevo s premerom 3 ali več. Ker ima premer vsaj 3 obstajata vozlišči u in v oddaljeni eno od druge za točno 3 preko najkrajše in edine poti, ki gre skozi dve točki, ki jih označimo z a in b. Tako imamo pot $v \to a \to b \to u$. Z s_v, s_a, s_b, s_u označimo število morebitnih točk poddreves upetih na v, a, b, u. Obravnavamo 2 možni zamenjavi: □



Lema 6.2. *V vsakem ravnovesnem grafu igre maksimalne razdalje se lokalni premer za katera koli* 2 *poljubna vozlišča razlikuje največ za* 1.

Dokaz. Predpostavimo, da je graf v *ravnotežju glede na maksimalno razdaljo* in ima vozlišče v z lokalnim premer d in vozlišče w z lokalnim premerom vsaj $d+2$. Naj bo T drevo, ki ga dobimo z iskanjem v širino iz vozlišča v. Vozlišče w z zamenjavo svoje povezave s staršem v T z povezavo do v (korena T) zmanjša svoj lokalni premer. Opazimo, da ta zamenjava lahko le zmanjša ali ohrani globine vozlišč v T, zato lokalni premer vozlišča v ostane največ d. Tako se lokalni premer vozlišča w zmanjša na vsaj $d+1$, saj se w lahko premakne po novo nastali povezavi wv do v in nato sledi poti v do vseh ostalih vozlišča. Ta zamenjava je v nasprotju s predpostavko, da je graf v *ravnotežju glede na maksimalno razdaljo* saj lahko vozlišče u izboljša svoj položaj(zmanjša svoj lokalni premer) z omenjeno zamenjavo povezav. □

Lema 6.3. *Če ima ravnovesni graf za maksimalno razdaljo prerezno vozlišče v, potem ima lahko samo ena izmed povezanih komponent $G-v$ vozlišče z razdaljo več kot 1 od v.*

Dokaz. Ponovno bomo dokazali lemo z protislovjem. Naj bo d lokalni premer prereznega vozlišča v in naj bo vozlišče u na razdalji d od v. Z U označimo povezano komponento $G-v$ ki vsebuje u. Predpostavimo, da $G-U$ vsebuje vozlišče z, ki je za več kot 1 oddaljeno od vozlišča v. Ker je vozlišče v prerezno in z in u nista vozlišči iste povezane komponente $G-v$, mora vsaka pot med njima prečkati v. Tako je najkrajša pot od z do u dolga $d+2$. Lokalni premer z in u je zato vsaj $d+2$ kar se za več kot 1 razlikuje od lokalnega premera vozlišča v in je tako v nasprotju z predhodno lemo in zato graf, ki ima več kot eno povezano komponento $G-v$ z vozliščem z razdaljo več kot 1 od v, ne more biti ravnovesni graf. □

Izrek 6.4. *Če je ravnovesni graf za maksimalno razdaljo drevo, potem ima premer največ 3.*

Dokaz. Predpostavimo, da je ravnovesni graf za maksimalno razdaljo drevo in ima premer vsaj 4. Potem obstajata vozlišči v in u, ki sta na razdalji točno 4 in med njima obstaja pot dolžine 4 $v \to a \to b \to c \to u$. Ker je graf drevo je vozlišče b prerezno vozlišče z dvema povezanima komponentama $G-b$, ki vsebujeta vozlišči v in u, ki sta na razdalji več kot 1 od b in tako v protislovju z predhodno lemo. □

Lema 6.5. *Za vozlišče v z lokalnim premerom 2, zamenjava sosednje povezave ne izboljša vsote razdalj od v do vseh ostalih vocližč. Prav tako tudi svojega lokalnega premera ne more izboljšati.*

Dokaz. Naj ima graf G n vozlišč in naj ima poljubno vozlišče z lokalnim premerom 2 stopnjo $deg(v) = k$. Tako ima vozlišče v, k sosednjih vozlišč in $n-k-1$ vozlišč na razdalji 2, saj je lokalni premer v enak 2. Vsota razdalj od v do vseh ostalih vozlišč je zato $1*k+2*(n-k-1)$. Vozlišče v z menjavo poljubne povezave ne spremeni števila sosednjih vozlišč k. Zamenjavo ene izmed povezav vozlišča v lahko tretiramo kot odstranitev ene obstoječe povezave, s katero se izgubi eno sosedno vozlišče, in dodajo nove povezave, s katero se pridobi eno sosednje vozlišče. Tako ima vozlišče v ne glede na zamenjavo povezav, k sosednjih vozlišč in $n-k-1$ vozlišč na razdalji vsaj 2, saj se razdalja do ne sosednjiega vozlišča lahko le poveča ali ostane enaka 2. Zato je vsota razdalj od v do vseh ostalih vocližč, po zamenjavah povezav v enaka ali večja od $1*k+2*(n-k-1)$. □

Posledica 6.6. *!!!!!!!!!! Vsak graf z premerom 2 ali manj je v ravnotežju glede na vsoto razdalj. In hkrati socialni optimum.*

Posledica 6.7. *Cena stabilnosti je 1.*

Izrek 6.8. *Naj bo G graf z n vozlišči in m povezavami, potem je $W(G) = n^2 - n - m$, če in samo če je premer grafa 2 ali manj.*

Dokaz. Predpostavimo, da je G graf reda n in velikosti m ter da velja $\text{diam}(G) \leq 2$. Definirajmo množici $A = \{u \in V | e(u) = 1\}$ in $B = \{u \in V | e(u) = 2\}$. Potem velja $|A| + |B| = n$. Če je $u \in A$, potem je $d(u) = n - 1$ (vsota vseh razdalj od u do ostalih vozlišč), in če je $u \in B$, definiramo dve množici B_1 in B_2 kot $B_1 = \{v \in V | d(u, v) = 1\}$ in $B_2 = \{v \in V | d(u, v) = 2\}$.

Nato velja:

$$\begin{aligned}
d(u) &= |B_1| + 2|B_2| \\
&= |B_1| + |B_2| + |B_2| \\
&= n - 1 + (n - 1 - |B_1|) \qquad \text{ker } |B_1| + |B_2| = n - 1 \\
&= 2n - 2 - |B_1| \\
&= 2n - 2 - \deg(u)
\end{aligned}$$

In zato sledi:

$$\begin{aligned}
W(G) &= \frac{1}{2} \sum_{u \in V} d(u) \\
&= \frac{1}{2}\Big(\sum_{u \in A} d(u) + \sum_{u \in B} d(u)\Big) \\
&= \frac{1}{2}\Big((n-1)|A| + \sum_{u \in B} 2n - 2 - \deg(u)\Big) \\
&= \frac{1}{2}\Big((n-1)|A| + (2n-2)|B| - \sum_{u \in B} \deg(u)\Big) \\
&= \frac{1}{2}\Big((n-1)(|A| + |B|) + (n-1)|B| - \sum_{u \in B} \deg(u)\Big) \\
&= \frac{1}{2}\Big((n-1)n + (n-1)(n - |A|) - \sum_{u \in B} \deg(u)\Big) \\
&= \frac{1}{2}\Big((n-1)n + (n-1)n - (n-1)|A| - \sum_{u \in B} \deg(u)\Big) \\
&= \frac{1}{2}\Big(2(n-1)n - \sum_{u \in A} \deg(u)\Big) - \sum_{u \in B} \deg(u)\Big) \\
&= \frac{1}{2}\Big(2(n-1)n - \sum_{u \in V} \deg(u)\Big) \\
&= \frac{1}{2}(2(n-1)n - 2m) \qquad \text{ker } \sum_{u \in V} \deg(u) = 2m, u \in V \\
&= n^2 - n - m
\end{aligned}$$

Kar dokaže, da je $W(G) = n^2 - n - m$ za grafe s premerom manjšim od 2 ($\operatorname{diam}(G) \leq 2$). V drugem delu bomo dokazali, da ta enakost ne velja za grafe z premerom večjim od 2. Za graf G ki ima premer vsaj 3 (($\operatorname{diam}(G) \geq 3$)) bomo definirali množici $A = \{u \in V | e(u) = 2\}$ in $B = \{u \in V | e(u) \geq 3\}$. Za kateri velja $|A| + |B| = n$. Če je $u \in A$, potem iz zgoraj dokazanega velja $d(u) = 2n - 2 - \deg(u)$. Za $u \in B$ pa bomo definirali sledeče 3 podmnožice:

$$\begin{aligned}
B_1 &= \{v \in V | d(u, v) = 1\},\\
B_2 &= \{v \in V | d(u, v) = 2\},\\
B_3 &= \{v \in V | d(u, v) \geq 3\}.
\end{aligned}$$

Očitno, $|B_1| + |B_2| + |B_3| = n - 1$.

$$\begin{aligned}
d(u) &\geq |B_1| + 2|B_2| + 3|B_3|\\
&= |B_1| + |B_2| + |B_3| + |B_2| + 2|B_3|\\
&= n - 1 + (n - 1 - |B_1|) + |B_3|\\
&= 2n - 2 - \deg(u) + |B_3|\\
&= 2n - 2 - \deg(u) + |B_3|\\
&\geq 2n - 2 - \deg(u) + 1 \qquad \text{saj } |B_3| \geq 1\\
&\geq 2n - 1 - \deg(u).\\
\therefore W(G) &= \frac{1}{2} \sum_{u \in V} d(u)\\
&= \frac{1}{2} (\sum_{u \in A} d(u) + \sum_{u \in B} d(u))\\
&\geq \frac{1}{2} ((\sum_{u \in A} (2n - 2 - \deg(u)) + (\sum_{u \in B} (2n - 1 - \deg(u)))\\
&= \frac{1}{2} ((2n - 2)(|A| + |B|) - \sum_{u \in A} \deg(u)) - (\sum_{u \in B} \deg(u)) + |B|)\\
&= \frac{1}{2} ((2n - 2)n - (\sum_{u \in V} \deg(u)) + |B|)\\
&= \frac{1}{2} (2(n - 1)n - 2m + |B|)\\
&= n(n - 1) - m + \frac{1}{2} |B|\\
&\geq n(n - 1) - m + 1 \qquad \text{saj } |B| \geq 2
\end{aligned}$$

□

Posledica 6.9. *Za vsak graf G z n vozlišči, m povezavami in premerom večjim od 2 velja:*

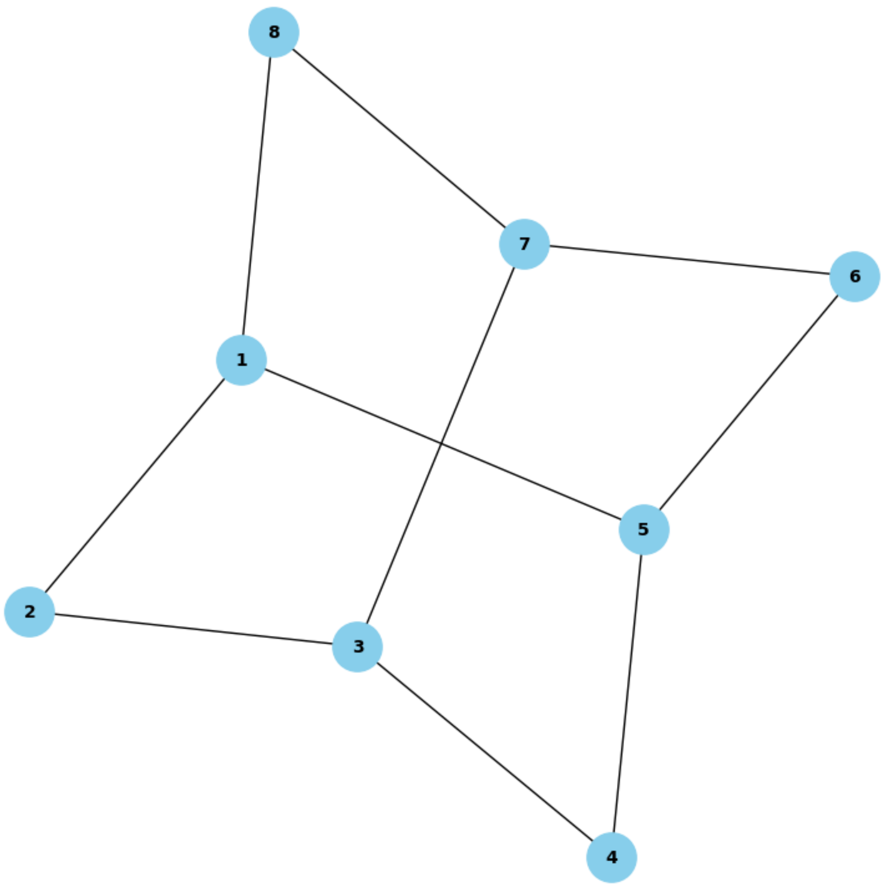
$$W(G) \geq n^2 - n - m + 1.$$

Enačaj velja kadar ima graf G natanko 2 vozlišča z lokalnim premerom 3 in vsa ostala vozlišča z lokalnim premerom 2.

Dokaz. □

Izrek 6.10. *Obstaja ravnovesni graf za skupno vsoto razdalj s premerom* 3.

Dokaz. Graf na spodnji sliki ima premer 3 in je v *ravnotežju glede na vsoto razdalj*. Točke 1, 3, 5 in 7 imajo lokalni premer 2 in tako po predhodni lemi ne morejo same zmanjšati svoje vsote razdalj do vseh ostalih vozlišč. Točke 2, 4, 6 in 8 so simetrične in celo povezavi teh točk so simetrične, zato je dovolj pogledati vse možne zamenjave le za eno izmed povezav teh točk. Točka 2 ima vsoto razdalj do vseh ostalih točk enako 1+1+2+2+2+2+3 = 13. Za preverjanje ali lahko točka 2 izboljša svoj položaj bomo poskusili z zamenjavo povezave med točkama 2 in 1 z novimi povezavami med točko 2 in ostalimi povezavami. Če povezavo 21 zamenjamo s povezavo 24 ima vozlišče 2 vsoto razdalj enako $1+1+2+2+3+3+3 = 15$. Če povezavo 21 zamenjamo s povezavo 25 ima vozlišče 2 vsoto razdalj enako $1+1+2+2+2+2+3 = 13$. Če povezavo 21 zamenjamo s povezavo 26 ima vozlišče 2 vsoto razdalj enako $1+1+2+2+2+3+3 = 14$. Če povezavo 21 zamenjamo s povezavo 27 ima vozlišče 2 vsoto razdalj enako $1+1+2+2+2+3+3 = 14$. Če povezavo 21 zamenjamo s povezavo 28 ima vozlišče 2 vsoto razdalj enako $1+1+2+2+2+3+3 = 13$. □

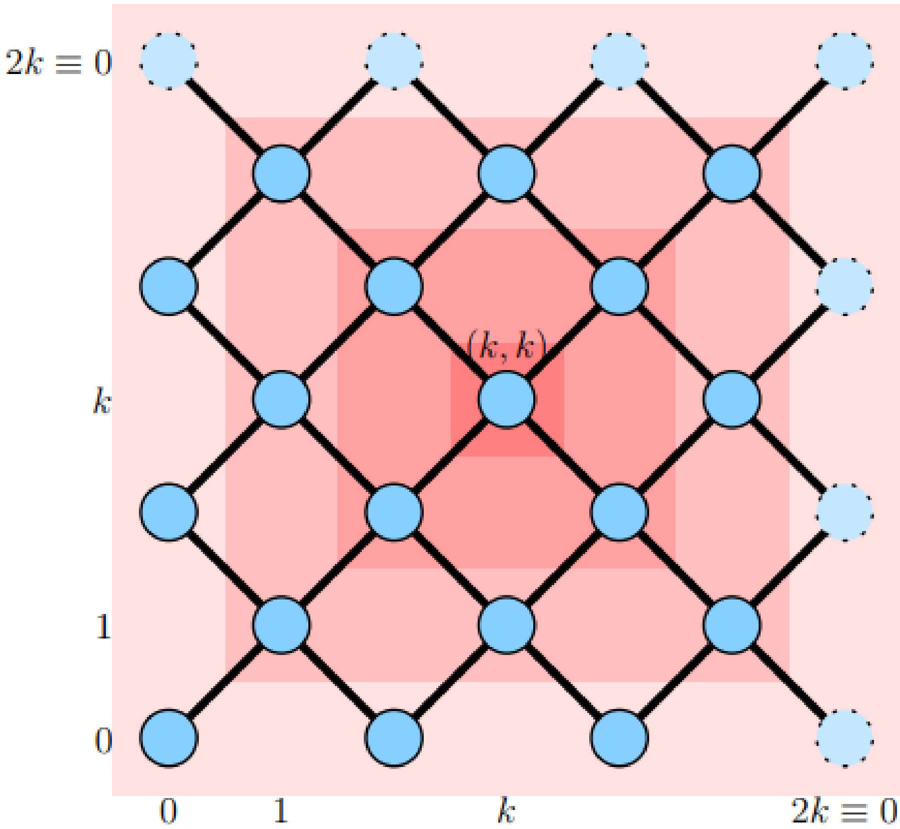


Izrek 6.11. *Vsak ravnovesni graf za skupno vsoto razdalj ima premer* $2^{O(\sqrt{\lg n})}$.

Lema 6.12. *Vsak ravnovesni graf za skupno vsoto razdalj ima premer največ* $2\lg n$ *ali za vsako vozlišče* v *obstaja povezava* xy *kjer je* $d(u,x) \leq \lg n$ *in zamenjava povezave* xy *zmanjša vsoto razdalj od* x *za največ* $2n(1+\lg n)$.

Lema 6.13. *V vsakem ravnovesnem grafu za skupno vsoto razdalj dodajanje poljubne povezave uv zmanjša vsoto razdalj od u za največ $5n \log n$.*

Izrek 6.14. *Obstaja ravnovesni graf za maksimalno razdaljo z premerom $\Theta(\sqrt{n})$.*



Izrek 6.15. *Vsak ravnovesni graf za skupno vsoto razdaljo G z $n \geq 24$ vozlišč in premerom $d > 2 \lg n$ inducira podgraf z ϵ-skoraj-enotno-razdaljo G' z n vozlišči in premerom $\Theta\left(\frac{\varepsilon d}{\lg n}\right)$ in podgraf z ϵ-enotno-razdaljo G' z n vozlišči in premerom $\Theta\left(\frac{\varepsilon d}{\lg^2 n}\right)$.*

Izrek 6.16. *Vsak ravnovesni graf za osnovne igre ustvarjanja omrežij (mogoče samo za vsoto) ima največ eno po povezavah 2−povezano komponento.*

Dokaz. !!!!!!!! □

7 Testiranje

V razdelku Testiranje smo s pomočjo programskega jezika python podobneje spoznali igre ustvarjanja omrežji, grafe, ki v takih igrah nastajajo in preverjali nekatere domneve o ravnovesnih grafih. Tako smo se prvo lotili primerjati število različnih ravnovesnih grafov med ravnovesnimi grafi glede na vsoto, ravnovesnimi grafi glede na razdaljo in grafi, ki so hkrati stabilni za dodajanje pvezav in kritični za odstranitev povezav.

Št. vozlišč	SUM	MAX	krit
2	1	1	1
3	2	2	1
4	5	3	2
5	15	4	3
6	60	7	5
7	374	12	8
8	4161	24	17

Tabela 1: Strnjeno po vozliščih

Po pričakovanjih smo opazili, da smo pri testiranju vseh grafov do 8 vozlišč opazili največ ravnovesnih glede na vsoto in najmanj kritičnih in stabilnih hkrati. Opazili smo tudi, da pri določenih kombinacijah števila vozlišč in števila povezav ravnovesni grafi glede na razdaljo ne obstajajo.

V nadaljevanju smo si ogledali nekaj različnih algoritmov za iskanje ravnovesnega grafa glede na največjo razdaljo in koliko od najdenih ravnovesnih grafov pri posameznem algoritmu je dreves.

Pogledali smo si tudi katera točka v drevesnih grafih pri igri doseže najboljšo pozicijo tako v igri kjer imajo vse povezave enako ceno in igri v realni ravnini.

Vozlišča	Povezave	SUM	MAX	Kritično
2	1	1	1	1
3	2	1	1	0
3	3	1	1	1
4	3	1	1	1
4	4	2	1	0
4	5	1	0	0
4	6	1	1	1
5	4	1	1	1
5	5	2	1	1
5	6	4	1	0
5	7	4	0	0
5	8	2	0	0
5	9	1	0	0
5	10	1	1	1
6	5	1	2	1
6	6	1	1	1
6	7	3	1	1
6	8	10	1	0
6	9	15	1	1
6	10	12	0	0
6	11	9	0	0
6	12	5	0	0
6	13	2	0	0
6	14	1	0	0
6	15	1	1	1
7	6	1	2	1
7	7	1	3	1
7	8	2	0	0
7	9	8	3	3
7	10	24	2	1
7	11	52	0	0
7	12	76	1	1

Vozlišča	Povezave	SUM	MAX	Kritično
7	13	75	0	0
7	14	57	0	0
7	15	38	0	0
7	16	21	0	0
7	17	10	0	0
7	18	5	0	0
7	19	2	0	0
7	20	1	0	0
7	21	1	1	1
8	7	1	3	1
8	8	1	3	2
8	9	2	1	1
8	10	7	4	2
8	11	15	4	3
8	12	55	4	3
8	13	165	2	2
8	14	387	0	0
8	15	649	1	1
8	16	787	1	1
8	17	733	0	0
8	18	567	0	0
8	19	370	0	0
8	20	211	0	0
8	21	111	0	0
8	22	56	0	0
8	23	24	0	0
8	24	11	0	0
8	25	5	0	0
8	26	2	0	0
8	27	1	0	0
8	28	1	1	1

Strnjeno:

Zanimal nas je tudi socialni optimum za max verzijo in ali je hkrati tudi ravnovesje.

Socialni optimum max

Premer grafa je največ 2. Vozlišča s premerom 2 bi si želela odstraniti nekatere povezave. Cena stabilnosti je lahko večja od 1 ampak manjša od 2. Ko gre n proti neskončno konvergira proti 2 ali 1. !!!!!! Optimalni grafi se stabilizirajo kot zvezde za max.

Socialni optimum sum, (opti, premer)

n\e	Število povezav															
	n - 1	**n**	**n + 1**	**n + 2**	**n + 3**	**n + 4**	**n + 5**	**n + 6**	**n + 7**	**n + 8**	**n + 9**	**n + 10**	**n + 11**	**n + 12**	**n + 13**	**n + 14**
2	(2, 'Je')															
3	(5, 'Je')	(3, 'Je')														
4	(7, 'Je')	(7, 'Ni')	(6, 'Ni')	(4, 'Je')												
5	(9, 'Je')	(9, 'Ni')	(9, 'Ni')	(8, 'Ni')	(8, 'Ni')	(7, 'Ni')	(5, 'Je')									
6	(11, 'Je')	(11, 'Ni')	(11, 'Ni')	(11, 'Ni')	(10, 'Ni')	(10, 'Ni')	(10, 'Ni')	(9, 'Ni')	(9, 'Ni')	(8, 'Ni')	(6, 'Je')					
7	(13, 'Je')	(13, 'Ni')	(13, 'Ni')	(13, 'Ni')	(13, 'Ni')	(12, 'Ni')	(12, 'Ni')	(12, 'Ni')	(12, 'Ni')	(11, 'Ni')	(11, 'Ni')	(11, 'Ni')	(10, 'Ni')	(10, 'Ni')	(9, 'Ni')	(7, 'Je')

Tabela 2: Your table caption goes here.

n\e	Število povezav															
	n - 1	**n**	**n + 1**	**n + 2**	**n + 3**	**n + 4**	**n + 5**	**n + 6**	**n + 7**	**n + 8**	**n + 9**	**n + 10**	**n + 11**	**n + 12**	**n + 13**	**n + 14**
2	(2, 1)															
3	(8, 2)	(6, 1)														
4	(18, 2)	(16, 2)	(14, 2)	(12, 1)												
5	(32, 2)	(30, 2)	(28, 2)	(26, 2)	(24, 2)	(22, 2)	(20, 1)									
6	(50, 2)	(48, 2)	(46, 2)	(44, 2)	(42, 2)	(40, 2)	(38, 2)	(36, 2)	(34, 2)	(32, 2)	(30, 1)					
7	(72, 2)	(70, 2)	(68, 2)	(66, 2)	(64, 2)	(62, 2)	(60, 2)	(58, 2)	(56, 2)	(54, 2)	(52, 2)	(50, 2)	(48, 2)	(46, 2)	(44, 2)	(42, 1)

Tabela 3: Your table caption goes here.

Primerjali smo nekaj različnih algoritmov za iskanje ravnovesja oziroma igranje igre in opazovali kateri algoritem ima večjo tenzijo ustvarit drevesa za ravnovesne grafe.

V tem besedilu bomo opisali štiri algoritme za optimizacijo grafov v latex kodi.

8 Algoritem: ekzaktno_max_1(graf)

Algoritem 1 išče optimalen graf tako, da za vozlišča po vrsti preveja eno po eno povezavo na način, da jo prvo poskuša odstraniti in ugotoviti ali je graf še vedno povezan in ali je vozlišče ki smo mu odstranili povezavo ohranilo lokalni premer. V kolikor odstranitev te povezave ni v interesu vozlišča poskuša odstranjeno povezavo zamenjati za še ne obstoječe povezave in preveri ali je graf povezan in se je lokalni premer vozlišča zmanjšal. Algoritem se ob vsaki spremembi grafa vrne na začetek in ponovno poizkuša odstraniti in zamenjati povezave od prvega vozlišča naprej. Algoriem vrne optimiziran graf, ko gre čez vsa vozlišča in njihove povezave brez, da bi se graf spremenil. Torej, ko nobeno vozlišče ne more več izboljšati svojega položaja.

Opis algoritmov

Algoritem 1

Algoritem 1 išče optimalen graf tako, da za vsako vozlišče iterativno pregleduje povezave. Postopek je naslednji:

1. **Odstranjevanje povezave**:
   - Za vsako vozlišče v grafu preveri vsako povezavo, ki izhaja iz tega vozlišča.
   - Poskuša odstraniti to povezavo in preveri, ali graf ostane povezan in ali lokalni premer (ekscentričnost) vozlišča ostane enak ali se zmanjša.
2. **Zamenjava povezave**:
   - Če odstranitev povezave ni koristna (graf postane nepovezan ali se ekscentričnost poveča), algoritem poskuša odstraniti to povezavo in dodati novo, še ne obstoječo povezavo z drugim vozliščem.
   - Preveri, ali graf ostane povezan in ali se ekscentričnost vozlišča zmanjša.
3. **Rekurzivnost**:
   - Če najde izboljšavo (graf ostane povezan in se ekscentričnost zmanjša), posodobi graf in se rekurzivno vrne na začetek ter ponovno preverja povezave od prvega vozlišča naprej.
4. **Stabilno stanje**:
   - Algoritem vrne optimiziran graf, ko gre čez vsa vozlišča in njihove povezave brez, da bi se graf spremenil. To pomeni, da nobeno vozlišče ne more več izboljšati svojega položaja.

Algoritem 2

Algoritem 2 je podoben algoritmu 1, vendar ima nekaj razlik v pristopu. Postopek je naslednji:

1. **Odstranjevanje in dodajanje povezave**:
   - Za vsako vozlišče v grafu preveri vsako povezavo, ki izhaja iz tega vozlišča.
   - Poskuša odstraniti to povezavo in dodati novo povezavo z drugim vozliščem ter preveri, ali graf ostane povezan in ali se lokalni premer (ekscentričnost) vozlišča zmanjša.
2. **Rekurzivnost**:
   - Če najde izboljšavo (graf ostane povezan in se ekscentričnost zmanjša), posodobi graf in se rekurzivno vrne na začetek ter ponovno preverja povezave od prvega vozlišča naprej.
3. **Stabilno stanje**:
   - Algoritem vrne optimiziran graf, ko gre čez vsa vozlišča in njihove povezave brez, da bi se graf spremenil. To pomeni, da nobeno vozlišče ne more več izboljšati svojega položaja.

Algoritem 3

Algoritem 3 prav tako išče optimalen graf s preverjanjem in spreminjanjem povezav, vendar s poudarkom na drugačnem zaporedju operacij. Postopek je naslednji:

1. **Odstranjevanje in dodajanje povezave**:
   - Za vsako vozlišče v grafu preveri vsako povezavo, ki izhaja iz tega vozlišča.
   - Poskuša odstraniti to povezavo in dodati novo povezavo z drugim vozliščem ter preveri, ali graf ostane povezan in ali se lokalni premer (ekscentričnost) vozlišča zmanjša.
2. **Dodatno preverjanje**:
   - Če nobena povezava ne izboljša stanja, ponovno preveri vse povezave in odstrani tiste, ki ne povečajo ekscentričnosti, če je graf še vedno povezan.
3. **Rekurzivnost**:
   - Če najde izboljšavo, posodobi graf in se rekurzivno vrne na začetek ter ponovno preverja povezave od prvega vozlišča naprej.
4. **Stabilno stanje**:
   - Algoritem vrne optimiziran graf, ko gre čez vsa vozlišča in njihove povezave brez, da bi se graf spremenil.

Algoritem 4

Algoritem 4 združuje tehnike iz prejšnjih algoritmov, a z dodatnimi preverjanji in koraki. Postopek je naslednji:

1. **Odstranjevanje in zamenjava povezave**:
   - Za vsako vozlišče v grafu preveri vsako povezavo, ki izhaja iz tega vozlišča.
   - Poskuša odstraniti to povezavo in jo nadomestiti z novo povezavo z drugim vozliščem ter preveri, ali graf ostane povezan in ali se ekscentričnost zmanjša.
2. **Dodatno preverjanje**:
   - Če nobena povezava ne izboljša stanja, ponovno preveri vse povezave in odstrani tiste, ki ne povečajo ekscentričnosti, če je graf še vedno povezan.
3. **Rekurzivnost**:
   - Če najde izboljšavo, posodobi graf in se rekurzivno vrne na začetek ter ponovno preverja povezave od prvega vozlišča naprej.
4. **Stabilno stanje**:
   - Algoritem vrne optimiziran graf, ko gre čez vsa vozlišča in njihove povezave brez, da bi se graf spremenil.

Algoritem `ekzaktno_max_1(graf)` optimizira graf tako, da išče ravnovesni graf z zmanjšanjem ekscentričnosti. Funkcija iterativno preverja vsako povezavo v grafu, odstrani povezavo, če odstranjena povezava ne vpliva na povezanost grafa in zmanjša ekscentričnost, ali doda novo povezavo, če zmanjša ekscentričnost grafa. Če nobena od pogojev ni izpolnjena, vrne izvirni graf kot ravnovesni graf.

9 Algoritem: ekzaktno_max_2(graf)

Algoritem `ekzaktno_max_2(graf)` optimizira graf s povečanjem povezav in zmanjšanjem ekscentričnosti. Funkcija iterativno preverja vsako povezavo v grafu, odstrani povezavo in doda novo, če nova povezava ne vpliva na povezanost grafa in zmanjša ekscentričnost. Prav tako odstrani povezavo, če ne vpliva na povezanost in zmanjša ali ohranja ekscentričnost grafa. Če nobena od pogojev ni izpolnjena, vrne izvirni graf kot ravnovesni graf.

10 Algoritem: ekzaktno_max_3(graf)

Algoritem `ekzaktno_max_3(graf)` optimizira graf s povečanjem povezav in zmanjšanjem ekscentričnosti. Funkcija iterativno preverja vsako povezavo v grafu, odstrani povezavo in doda novo, če nova povezava ne vpliva na povezanost grafa in zmanjša ekscentričnost. Nato ponovno preveri vse povezave in odstrani povezavo, če ne vpliva na povezanost in zmanjša ali ohranja ekscentričnost grafa. Če nobena od pogojev ni izpolnjena, vrne izvirni graf kot ravnovesni graf.

11 Algoritem: ekzaktno_max_4(graf)

Algoritem `ekzaktno_max_4(graf)` optimizira graf z zmanjšanjem ekscentričnosti s pomočjo dveh faz. V prvi fazi funkcija iterativno odstranjuje povezave in preverja povezanost grafa ter ekscentričnost. Če je ekscentričnost manjša ali enaka, graf shrani kot ravnovesni graf in ponovno zažene optimizacijo. V drugi fazi algoritem odstranjuje povezave in dodaja nove, če nova povezava zmanjša ekscentričnost in ohranja povezanost. Če nobena od pogojev ni izpolnjena, vrne izvirni graf kot ravnovesni graf.

Algoritem začne z začetnim grafom in pokliče rekurzivno funkcijo loop$_max.Zavsakovozli$šč$ev1preverjanjegovepovezave,kjerposku$š$ab$

Optimalno pozicioniranje na tržišču.

Pri igri vsoto smo uporabili dva podobna algritma za iskanje ravnovesja. Oba algoritma delujeta tako, da prva točka v grafu optimizira svoj položaj glede na stanje ostalih točk nato druga točka po vrsti optimizira svoj položaj. V kolikor se je graf spremenil ponono prva točka optimizira svoj položaj glede na novo nastalo stanje, sicer pa je na vrsti tretja točka. Torej točke po vrsti optimizirajo svoj položaj in ob vsaki spremembi grafa se vrstni red ponovno začne pri prvi točki. Algoritma se razlikujeta le po temu, da eden izmed algoritmov poišče ravnovesni graf za igro vsote za grafe v prostoru (cena povezave je evklidska razdalja med točkami) drugi pa za grafe, kjer je cena vsake povezave enaka 1.

Algoritme smo testirali na 100000 naključno generiranih grafih za drevesne grafe z vozlišči od 3 do 10. Zanimalo nas je s kakšno verjetnostjo zmaga graf glede na svojo pozicijo v vrstnem redu igranja, glede na začetno število povezav, glede na začetno ceno in glede na centralnost pozicije v prostoru.

Iz teorije že vemo, da bo pri grafih kjer so vse povezave enake 1 le en zmagovalec(z najcenejšo ceno med vozlišči) saj bo graf zvezda. Testiranje za grafe v prostoru pa je pokazalo, da je pri vseh grafih z lihim številom vozlišč po le en zmagovalec (z najcenejšo ceno med vozlišči). Za grafe s sodimi vozlišči pa je situacija nekoliko drugačna. Za grafe z 10 vozlišči smo dobili 113222 zmagovalcev pri 100000 naključnih grafih. Za grafe z 8 vozlišči smo dobili 116603 zmagovalcev. Za grafe z 6 vozlišči smo dobili 125978 zmagovalcev. Za grafe z 4 vozlišči smo dobili 154551 zmagovalcev.

Iz teorije prav tako vemo, da bo pri grafih kjer so vse povezave enake 1 naš edini zmagovalec tudi imela največ povezav in ne le najnižjo ceno. Testiranje za grafe v prostoru pa je pokazalo, da imamo za grafe z 10 vozlišči 94620 zmagovalcev, ki imajo tudi največ povezav v ravnovesnem grafu, to je približno $\frac{94620}{113233} * 100 = 83,562\%$. Za grafe z 9 vozlišči imamo 89655 takšnih zmagovalcev, to je $\frac{89655}{100000} * 100 = 89,655\%$. Za grafe z 8 vozlišči imamo 100732 takšnih zmagovalcev, to je približno $\frac{100732}{116603} * 100 = 86,389\%$. Za grafe z 7 vozlišči imamo 93568 takšnih zmagovalcev, to je $\frac{93568}{100000} * 100 = 93,568\%$. Za grafe z 6 vozlišči imamo 112051 takšnih zmagovalcev, to je približno $\frac{112051}{125978} * 100 = 83,56\%$. Za grafe z 5 vozlišči imamo 100000 takšnih zmagovalcev, to je $\frac{100000}{100000} * 100 = 100\%$. Za grafe z 4 vozlišči imamo 154551 takšnih zmagovalcev, to je približno $\frac{154551}{154551} * 100 = 100\%$. Za grafe z 3 vozlišči imamo 100000 takšnih zmagovalcev, to je $\frac{100000}{100000} * 100 = 100\%$.

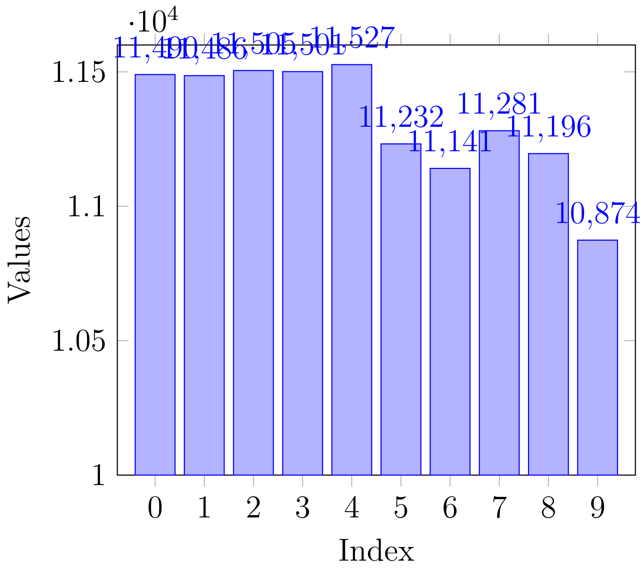
Zanimalo nas je tudi kako pogosto so vozlišča z največ povezavami na začetku postala zmagovalna vozlišča. Testiranje za grafe kjer so vse povezave enake 1 je dalo sledeče rezultate. Za grafe z 10 vozlišči je 72313 vozlišč zmagalo po tem, ko je imelo na začetku največje število povezav, to je približno $\frac{72313}{100000} * 100 = 72,313\%$. Za grafe z 9 vozlišči imamo 72841 takšnih vozlišč, to je $\frac{72841}{100000} * 100 = 72,841\%$. Za grafe z 8 vozlišči imamo 76813 takšnih vozlišč, to je $\frac{76813}{100000} * 100 = 76,813\%$. Za grafe z 7 vozlišči imamo 80178 takšnih vozlišč, to je $\frac{80178}{100000} * 100 = 80,178\%$. Za grafe z 6 vozlišči imamo 86408 takšnih vozlišč, to je $\frac{86408}{100000} * 100 = 86,408\%$. Za grafe z 5 vozlišči imamo 100000 takšnih vozlišč, to je $\frac{100000}{100000} * 100 = 100\%$. Za grafe z 4 vozlišči imamo 100000 takšnih vozlišč, to je $\frac{100000}{100000} * 100 = 100\%$. Za grafe z 3 vozlišči imamo 100000 takšnih vozlišč, to je $\frac{100000}{100000} * 100 = 100\%$.

Testiranje za grafe v prostoru pa je podalo sledeče rezultate. Za grafe z 10 vozlišči je 21984 vozlišč zmagalo po tem, ko je imelo na začetku največje število povezav, to je približno $\frac{21984}{113233} * 100 = 19,415\%$. Za grafe z 9 vozlišči imamo 21102 takšnih vozlišč, to je $\frac{21102}{100000} * 100 = 21,102\%$. Za grafe z 8 vozlišči imamo 27172 takšnih vozlišč, to je približno $\frac{27172}{116603} * 100 = 23,303\%$. Za grafe z 7 vozlišči imamo 27380 takšnih vozlišč, to je $\frac{27380}{100000} * 100 = 27,380\%$. Za grafe z 6 vozlišči imamo 41503 takšnih vozlišč, to je približno $\frac{41503}{125978} * 100 = 32,945\%$. Za grafe z 5 vozlišči imamo 40602 takšnih vozlišč, to je $\frac{40602}{100000} * 100 = 40,602\%$. Za grafe z 4 vozlišči imamo 68532 takšnih vozlišč, to je približno $\frac{68532}{154551} * 100 = 44,343\%$. Za grafe z 3 vozlišči imamo 33533 takšnih vozlišč, to je $\frac{33533}{100000} * 100 = 33,533\%$.

Zanimalo nas je tudi kako pogosto so vozlišča z najmanjšo začetno ceno postala zmagovalna vozlišča. Testiranje za grafe kjer so vse povezave enake 1 je dalo sledeče rezultate. Za grafe z 10 vozlišči je 70317 vozlišč zmagalo po tem, ko je imelo na začetku največje število povezav, to je približno $\frac{70317}{100000} * 100 = 72,313\%$. Za grafe z 9 vozlišči imamo 54692 takšnih vozlišč, to je $\frac{54692}{100000} * 100 = 54,692\%$. Za grafe z 8 vozlišči imamo 76225 takšnih vozlišč, to je $\frac{76225}{100000} * 100 = 76,225\%$. Za grafe z 7 vozlišči imamo 57105 takšnih vozlišč, to je $\frac{57105}{100000} * 100 = 57,105\%$. Za grafe z 6 vozlišči imamo 87578 takšnih vozlišč, to je $\frac{87578}{100000} * 100 = 87,578\%$. Za grafe z 5 vozlišči imamo 66467 takšnih vozlišč, to je $\frac{66467}{100000} * 100 = 66,467\%$. Za grafe z 4 vozlišči imamo 100000 takšnih vozlišč, to je $\frac{100000}{100000} * 100 = 100\%$. Za grafe z 3 vozlišči imamo 100000 takšnih vozlišč, to je $\frac{100000}{100000} * 100 = 100\%$.

Testiranje za grafe v prostoru pa je podalo sledeče rezultate. Za grafe z 10 vozlišči je 15780 vozlišč zmagalo po tem, ko je imelo na začetku največje število povezav, to je približno $\frac{15780}{113233} * 100 = 13,936\%$. Za grafe z 9 vozlišči imamo 12450 takšnih vozlišč, to je $\frac{12450}{100000} * 100 = 12,450\%$. Za grafe z 8 vozlišči imamo 20827 takšnih vozlišč, to je približno $\frac{20827}{116603} * 100 = 17,861\%$. Za grafe z 7 vozlišči imamo 15407 takšnih vozlišč, to je $\frac{15407}{100000} * 100 = 15,407\%$. Za grafe z 6 vozlišči imamo 31393 takšnih vozlišč, to je približno $\frac{31393}{125978} * 100 = 24,919\%$. Za grafe z 5 vozlišči imamo 20855 takšnih vozlišč, to je $\frac{20855}{100000} * 100 = 20,855\%$. Za grafe z 4 vozlišči imamo 68532 takšnih vozlišč, to je približno $\frac{68532}{154551} * 100 = 44,343\%$. Za grafe z 3 vozlišči imamo 33533 takšnih vozlišč, to je $\frac{33533}{100000} * 100 = 33,533\%$.

Zanimalo nas je tudi kako pogosto so vozlišča z najmanjšo možno končno ceno postala zmagovalna vozlišča. Za grafe kjer so vse povezave enake 1 so vsa vozlišča takšna saj imajo vsa teoretično možnost imeti strošek $n - 1$. Testiranje za grafe v prostoru pa je podalo sledeče rezultate. Za grafe z 10 vozlišči je zmagalo 76295 vozlišč, ki so imela najmanjšo možno končno ceno, to je približno $\frac{76295}{113233} * 100 = 67,379\%$. Za grafe z 9 vozlišči imamo 74767 takšnih vozlišč, to je $\frac{74767}{100000} * 100 = 74,767\%$. Za grafe z 8 vozlišči imamo 80006 takšnih vozlišč, to je približno $\frac{80006}{116603} * 100 = 68,614\%$. Za grafe z 7 vozlišči imamo 78229 takšnih vozlišč, to je $\frac{78229}{100000} * 100 = 78,229\%$. Za grafe z 6 vozlišči imamo 85553 takšnih vozlišč, to je približno $\frac{85553}{125978} * 100 = 67,911\%$. Za grafe z 5 vozlišči imamo 85374 takšnih vozlišč, to je $\frac{85374}{100000} * 100 = 85,374\%$. Za grafe z 4 vozlišči imamo 96570 takšnih vozlišč, to je približno $\frac{96570}{154551} * 100 = 62,484\%$. Za grafe z 3 vozlišči imamo 99888 takšnih vozlišč, to je $\frac{99888}{100000} * 100 = 99,888\%$.



Slika 1: Histogram of Given Data

12 ...

13 Zaključek

Slovar strokovnih izrazov