

1 Uvod

V zaključni nalogi pri predmetu Modelska Analiza I, raziskujemo model razporejanja prenosa podatkov po omrežju ([Naloga 16](#)). Model omrežja, ki ga obravnavamo je zastavljen kot:

Omrežje sestavlja $N \times N$ kvadratna mreža, kjer vsak rob predstavlja povezavo z naključno hitrostjo med $[0, 1]$. Vozlišča na zgornjem robu predstavljajo internetne odjemalce, na spodnjem robu pa strežnike. V notranjih vozliščih velja prvi Kirchhoffov zakon.

Analizirali bomo lastnosti modela, ko je skupna hitrost prenosa največja.

2 Formalizacija modela

Začnimo z definicijo usmerjenega grafa G .

Definicija 2.1. Usmerjen graf je urejen par $G = (V, E)$, kjer je V končna množica vozlišč, $E \subseteq V \times V$ pa množica usmerjenih povezav. Če je $(u, v) \in E$, pravimo, da obstaja usmerjena povezava iz vozlišča u v vozlišče v .

Omrežje, ki ga sestavlja $N \times N$ kvadratna mreža je primer usmerjenega grafa iz G definicije 2.1. Množica vozlišč je določena kot

$$V = \{(i, j) \in (\mathbb{N}, \mathbb{N}) \mid 1 \leq i \leq N, 1 \leq j \leq N\}. \quad (1)$$

Vozlišča, ki predstavljajo **strežnike** in **odjemalce** označimo s $s_j = (N, j) \in V$ in $o_j = (1, j) \in V$. Povezave med sosedji opišemo z uporabo manhatanske razdalje ($\|\cdot\|_1$),

$$E = \{\{(i, j), (k, l)\} \mid (i, j), (k, l) \in V \wedge \neg((i = k = N) \vee (i = k = 1)), \|(i - k, j - l)\|_1 = 1\}. \quad (2)$$

Pogoj, $(i, j), (k, l) \in V \wedge \neg((i = k = N) \vee (i = k = 1))$, izhaja iz izbrane interpretacije naloge, kjer prevzamemo, da povezave med strežniki ali med odjemalci niso dovoljene.

Definicija 2.2. Kapaciteta povezave je preslikava $c : E \rightarrow [0, 1]$, ki vsaki povezavi $(u, v) \in E$ priedi naključno vrednost $c_{uv} \in [0, 1]$. Preslikava c je simetrična na zamenjavo vozlišč v povezavi $(u, v) \in E$:

$$c_{uv} = c_{vu}. \quad (3)$$

Definicija 2.2 je prilagojena nalogi, ki jo obravnavamo. V standardni literaturi se kapaciteto povezave uporablja bolj splošno ([problem maksimalnega pretoka](#)).

Definicija 2.3. Pretok grafa je preslikva $f : E \rightarrow \mathbb{R}^+$, ki vsaki povezavi $(u, v) \in E$ priedi vrednost $f_{uv} \leq c_{uv}$ in upošteva prvi Kirchhoffov zakon:

$$\forall v \in V \setminus \{u \in V \mid \exists i, u = s_i \vee u = o_i\} : \sum_{u:(u,v) \in E} f_{uv} - \sum_{w:(v,w) \in E} f_{vw} = 0. \quad (4)$$

Cilj naloge je obravnavati različne količine omrežja, ko ima ta največjo skupno hitrost prenosa.

Skupno hitrost prenosa označimo z $|f|$, ki jo določimo na podlagi pretoka f :

$$|f| = \sum_i \sum_{u:(s_i, u) \in E} f_{s_i u} = \sum_j \sum_{w:(w, o_j) \in E} f_{w o_j}. \quad (5)$$

V nadaljevanju uporabimo standardni pristop, pri katerem z enosmernimi povezavami povežemo *Superstrežnik* S z vsemi strežniki s_i , ter vse odjemalce o_j s *Superodjemalcem* O . Kapacitete teh novih povezav niso omejene:

$$\forall i, \quad c_{Ss_i} = \infty, \quad c_{o_i O} = \infty. \quad (6)$$

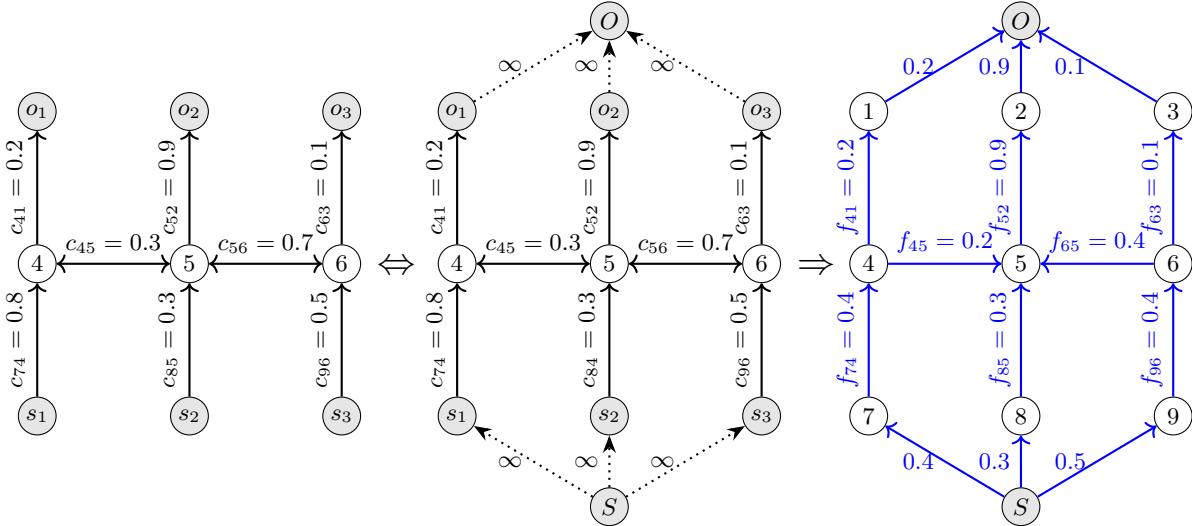
Na ta način strežnike in odjemalce obravnavamo kot običajna vozlišča v omrežju. Največjo skupno hitrost prenosa izrazimo kot linearen program:

$$\begin{aligned} \max \quad & |f| = \sum_{v:(S,v) \in E} f_{Sv} = \sum_{w:(w,O) \in E} f_{wO} \\ \text{o.p.} \quad & \sum_{u:(u,v) \in E} f_{uv} = \sum_{w:(v,w) \in E} f_{vw} \quad \forall v \in V \setminus \{S, O\} \\ & f_{uv} \leq c_{uv} \quad \forall (u, v) \in E \\ & f_{uv} \geq 0 \quad \forall (u, v) \in E \end{aligned} \quad (7)$$

Z linearimi programi smo smo se že srečali v 2. nalogi pri predmetu *Modelska analiza 1*, kjer smo jih reševali z znano [Simplekтиčno metodo](#).

2.1 Primer

Da bi bolje razumeli zgoraj formalno opisan model si oglejmo primer omrežja velikosti 3×3 . Na **Sliki 1** je prikazano omrežje in rešitev maksimalnega pretoka za dano razdelitev kapacitet.



Slika 1: Model omrežja predstavljen kot usmerjen graf z dodeljenimi kapaciteti na povezavah. Skrajno levi graf prikazuje osnovni model omrežja, kjer obojesmerne povezave omogočajo pretok podatkov v obe smeri. Sredinski graf je pridobljen iz levega tako, da sta dodana *Superstrežnik* in *Superodjemalec*, ki sta povezana z strežniki in odjemalci preko povezav z neomejeno kapaciteto. Skrajno desni graf prikazuje rešen problem maksimalnega pretoka za dan primer. Rezultat maksimalnega pretoka je $|f| = 0.2 + 0.9 + 0.1 = 1.2$.

Na **Sliki 1** so vozlišča oštrevlčena vrstično od leve proti desni in od zgoraj navzdol, kar ni skladno z opisom v enačbi (1). Razlog za to bomo pojasnili v Poglavlju 3. Rezultat maksimalnega pretoka ($|f| = 1.2$) na skrajno desnem grafu je zagotovo pravilen, saj je pretok do odjemalcev dosegel kapacitete teh povezav. Na tej točki ni povsem jasno kako smo lahko v splošnem prepričani, da je pretok maksimalen.

2.2 Dualni problem

Pri določanju optimalnosti rešitve linearne naloge si pomagamo z dualnim problemom.

1 Linearna naloga

Prvotno linearno nalogo in njen dual zapišemo kot:

Prvotna linearna naloga

$$\max \mathbf{c}^T \mathbf{x}$$

Pri pogojih:

$$\mathbf{A}\mathbf{x} \leq \mathbf{b}, \quad \mathbf{x} \geq \mathbf{0}$$

Dualna linearna naloga

$$\min \mathbf{b}^T \mathbf{y}$$

$$\mathbf{A}^T \mathbf{y} \geq \mathbf{c}, \quad \mathbf{y} \geq \mathbf{0}$$

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix}$$

Prvotni in dualni problem sta dve različni izražavi istega optimizacijskega problema. V prvotnem problemu iščemo vektor \mathbf{x} , ki maksimizira ciljno funkcijo ob danih omejitvah. V dualnem problemu pa iščemo vektor \mathbf{y} , ki minimizira skupno vrednost omejitev, tako da zagotovimo, da je vrednost ciljne funkcije vsaj enaka ali večja od tiste, ki jo določa prvotni problem. Velja izrek [1]:

Izrek 2.1 (Izrek o močni dualnosti). Če ima bodisi prvotni (P) bodisi dualni (D) linearni program končno optimalno vrednost, potem jo ima tudi drugi. Optimalni vrednosti obeh problemov sovpadata, obenem pa obstajajo optimalne rešitve tako za prvotni kot za dualni problem.

Pri problemu maksimalnega pretoka iščemo največjo količino pretoka, ki jo lahko prenesemo iz *Superstrežnika* do *Superodjemalca* skozi omrežje ob upoštevanju kapacitet povezav. Dualni problem išče najmanjšo skupno kapaciteto tistih povezav, katerih odstranitev bi popolnoma prekinila povezavo med začetnim in končnim vozliščem. Dualni problem imenujemu tudi problem najmanšega prereza. Iz izreka 2.1 sledi, da je maksimalen pretok dosežen natanko tedaj, ko obstaja minimalen prerez grafa, katerega skupna vrednost kapacitet je enaka velikosti pretoka.

Teoretičnen napor ni bil zamanj, saj imamo sedaj globji vpogled v to kaj omejuje velikost pretoka. Dodatno tudi vidimo, da je rešitev pretoka na **Sliki 1** maksimalna, saj je vrednost prereza povezav do strežnikov med prvo in drugo vrstico enaka velikosti maksimalnega pretoka.

3 Implementacija in reševanje naloge

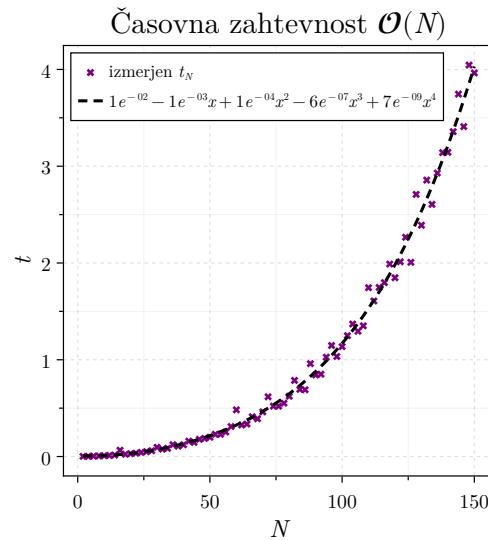
Model omrežja sem najprej implementiral v programskejem jeziku *Julia* [2]. Za prepis modela v linearen program sem uporabil knjižnico *JuMP* [3]. *JuMP* poenostavi zapis linearne naloge, saj mu je potrebno podati le matriko kapacitet, kjer pozicija (i, j) opisuje kapaciteto med vozliščem i in j . V primeru, da želimo obojesmerne povezave je matrika kapacitet simetrična. Linearen program smo reševali s pomočjo odprtokodnega solverja *HiGHS* [4]. Problem uporabe matrike kapacitet je njena velikost $N^2 \times N^2$, kar prestavlja N^4 povezav, kjer jim ima večina kapaciteto 0. Kljub temu, da je *HiGHS* ”pameten” in zanemari veliko povezav, se problema ni dalo rešiti za $N > 20$. Jasno je da število povezav v odvisnosti od N ni N^4 ampak približno $8N^2$ (odvisno kakšno usmerjenost prevzamemo). *JuMP* omogoča vnos vozlišč in njihovih kapacitet kot nabor $(u, v, c(u, v))$. Ker so u in v vozlišča označena z naravnimi števili smo kordinati (i, j) na mreži priredili število

$$(i, j) \rightarrow i \cdot N + j \quad (8)$$

S tem smo priredili koordinatam vozlišča v množici $u, v \in (0, 1, 2, \dots, N^2 - 1)$. Izbrane oznake osmislijo številčenje vozlišč na **Sliki 1**. Nov način implementacije se je izkazal za boljšega, saj smo v zglednem času rešili primer za $N = 100$.

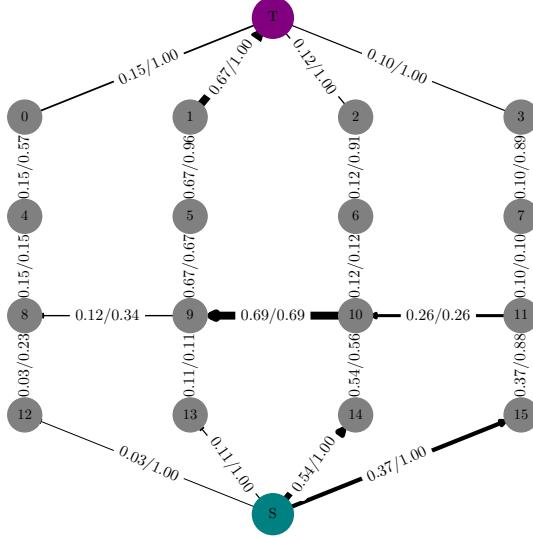
```
julia> @time test(100)
Running HiGHS 1.11.0 (git hash: 364c83a51e): Copyright (c) 2025 HiGHS under MIT licence terms
LP has 49801 rows; 39800 cols; 119400 nonzeros
Summary
    Runtime: 0.64s
Model status : Optimal
HiGHS run time : 0.80
objective_value(model) = 30.23404120194106
julia> 1.139361 seconds (5.17 M allocations: 246.883 MiB, 4.16% gc time)
```

Algoritmom uporabljen pri reševanju linearnega problema se imenuje *Interior point method*. **Slika 2** prikazuje časovno zahtevnost reševanja omrežja v odvisnosti od N . Zapis e^0 označuje potenco 10^0 . Čas na grafu je v sekundah. Prilagojen polinom stopnje 4 dobro opiše obnašanje na tej skali, vendar iz dodatnih testov pokaže od-sopanja. Za primer $N = 500$ potrebuje metoda $t_{500} \approx 250$ s. Smiselno je predpostaviti, da ima prilagojen polinom višje člene, ki so na tej skali zanemarljivi. Preverimo sedaj nekaj rešitev modela omrežja za majhne vrednosti N . Na **Sliki 3** sta prikazani rešitvi omrežja za različne kapacitete. Kapacitete so tukaj enakomerno porazdeljene na intervalu $[0, 1]$. Kot pričakovano opazimo, da je rezultat precej odvisen od porazdelitve kapacitet. Na **Sliki 4** sem dodal tudi minimalni prerez omrežij, da se lahko prepričamo v pravilnost rezultatov.

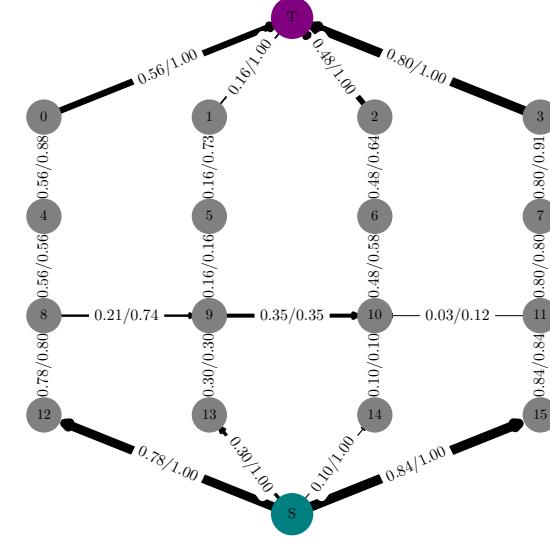


Slika 2: Prikaz časovne zahtevnosti reševanja maksimalnega toka za mrežo velikosti ($N \times N$).

N=4, $|f| = 1.05$ (Run 0)

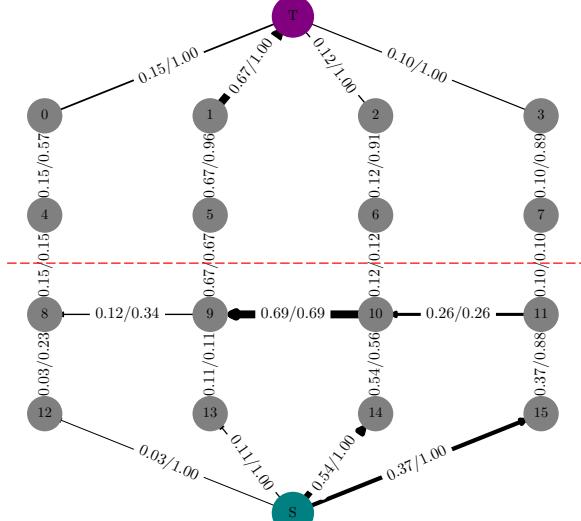


N=4, $|f| = 2.01$ (Run 1)

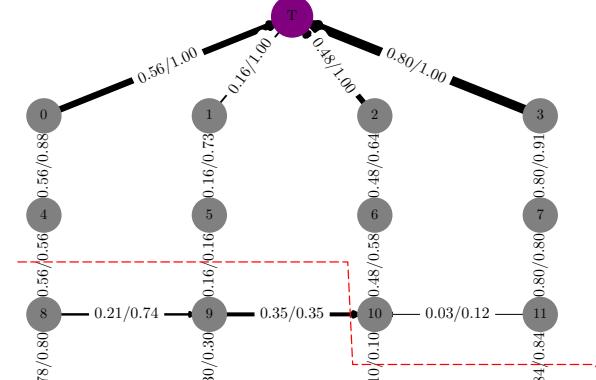


Slika 3: Prikaz dveh rešitev maksimalnega pretoka pri velikosti omrežja (4×4). Zaradi naključnosti kapacitet se rešitvi razlikujeta. Za rešitvi tudi velja ohranitev tokov v notranjih vozliščih.

N=4, $|f| = 1.05$ (Run 0)
 $0.15+0.67+0.12+0.1 \approx 1.05$



N=4, $|f| = 2.01$ (Run 1)
 $0.56+0.16+0.35+0.1+0.84 = 2.01$



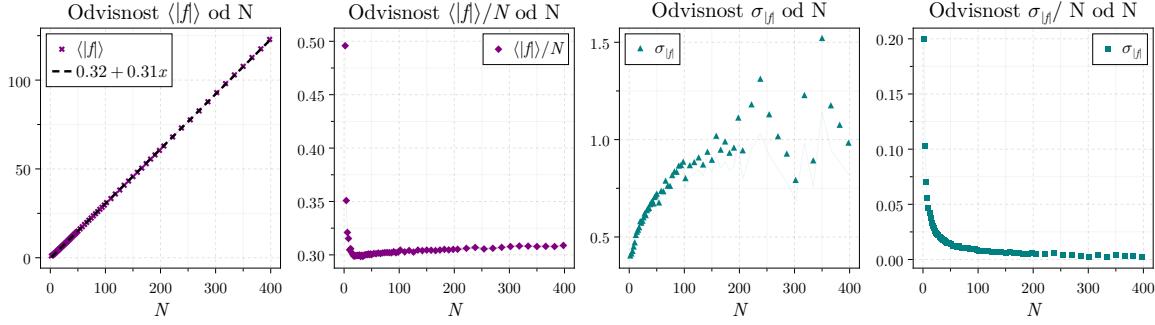
Slika 4: Rešitvi maskimalnega pretoka omrežja velikosti (4×4). Z rdečo črtko označimo minimalni prerez. Vidimo da minimalni prerez in velikost pretoka sovpadata, zato je rešitev res ekstremna.

Ker je omrežje naključno, nas bodo zanimala ansambelska povprečja. Dodatno si mislimo, da so rezultati zanimivejši v limiti $N \rightarrow \infty$. Problem, ki ga imamo je, da je linearno programiranje počasno za $N > 100$ tudi če paralelno računamo (imam le 8 threadov). Problem je hitreje rešljiv v primeru, da uporabljamo znane metode reševanja za **problem maksimalnega pretoka**. Ravno za maksimalni pretok je bil razvit znan algoritem **Ford-Fulkerson**. V jeziku *c++* sem iz knjižnice *LEMON* [5] uporabil **Push-relabel maximum flow algorithm**. Primerjano hitrost reševanja maksimalnega pretoka med linearnim programiranjem in algoritmom preflow (Push-relabel) prikazuje **Slika 5**. Opazimo, da je kompleksnost podobne polinomske stopnje, kljub temu je reševanje v *c++* s knjižnico *LEMON* za en velikostni red hitrejše. Opazimo tudi, da je možno rešiti maksimalni pretok omrežja velikosti (1000×1000).

```
./solve_gridgraph 1000 1 ".../networks" 0.5 1000 258.69s user 0.11s system 100% cpu
```

4 Analiza omrežja

Implementacija omrežja v *c++* nam omogoča lažjo analizo lastnosti omrežja. Prva lastnost, ki si jo bomo ogledali je odvisnost maksimalnega pretoka od parametra N . Preden analiziramo rezultate premislimo teoretične meje maksimalnega pretoka omrežja. Zamislimo si, da imajo vse povezave kapaciteto $c = 1$. V tem primeru je maksimalni pretok enak N , saj je to dolžina najkrajšega prereza. Na **Sliki 6** so prikazani rezultati analize maksimalnega pretoka v odvisnosti od N .

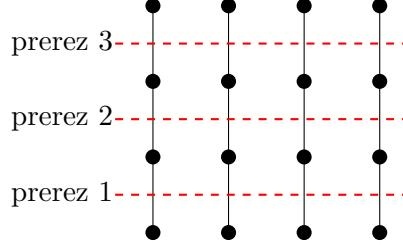


Slika 6: Analiza maksimalnega pretoka glede na velikost omrežja ($N \times N$). Rezultati prikazujejo linearno odvisnost med $|f|$ in N . Vsa povprečja so izračunana na ansamblu velikosti 1000, razen za $N > 300$, kjer je bila uporabljenha velikost ansambla 100.

Pričakovana vrednost pretoka narašča linearno z velikostjo sistema N , pri čemer je naklon premice $k \approx 0.3$. Standardni odklon prav tako narašča z N , pri čemer oblika rasti spominja na korenko odvisnost. Razmerje $\frac{|f|}{N}$ sicer ni povsem konstantno, temveč zmersno narašča. Izmerjene lastnosti

Slika 5: Primerjava časovnih zahtevnosti linearnega programiranja preflow algoritma. Graf je risan v log-log skali.

omrežja skušamo razložiti na sledeč način. Izpeljimo zgornjo mejo rasti maksimalnega pretoka. Pomagamo si z znanjem o dualnem problemu. Najkrajši prerez omrežja (ne nujno minimalen prerez) ima dolžino N . Predpostavimo, da je omrežje povezano le z vertikalnimi povezavami. To pomeni, da lahko omrežje prerezemo na $N - 1$ prerezov velikosti N . Primer takšnega omrežja veliko (4×4) prikazuje **Slika 7**. Ker je minimalni prerez manjši od vseh mora biti tudi manjši od minima



Slika 7: Preprost primer (4×4) omrežja horizontalnih povezav. Vse možne prerezne označimo s rdečo črtkano črto.

izbranih $N - 1$ prerezov. Po izreku o šibki dualnosti bo ta vrednost zagotovo večja od vrednosti maksimalnega pretoka. Izpeljimo porazdelitev minima opisanih $N - 1$ prerezov. Označimo i -ti horizontalni prerez H_i . Izrazimo H_i :

$$H_i = \sum_{j=1}^N c_{ij}, \quad \text{kjer} \quad c_{ij} \sim U(0, 1). \quad (9)$$

H_i je porazdeljeno kot vsota N neodvisnih enakomerno porazdeljenih naključnih spremenljivk $c_{i,j}$. Naj bo $|f|_{\max}$ minimum med izbranimi horizontalnimi prerezimi:

$$|f|_{\max} = \min\{H_1, H_2, \dots, H_{N-1}\}. \quad (10)$$

Kumulativna verjetnost $|f|_{\max}$ je podana kot:

$$\mathcal{P}(\min\{H_1, H_2, \dots, H_{N-1}\} \leq x) = 1 - (\mathcal{P}(H > x))^{N-1}. \quad (11)$$

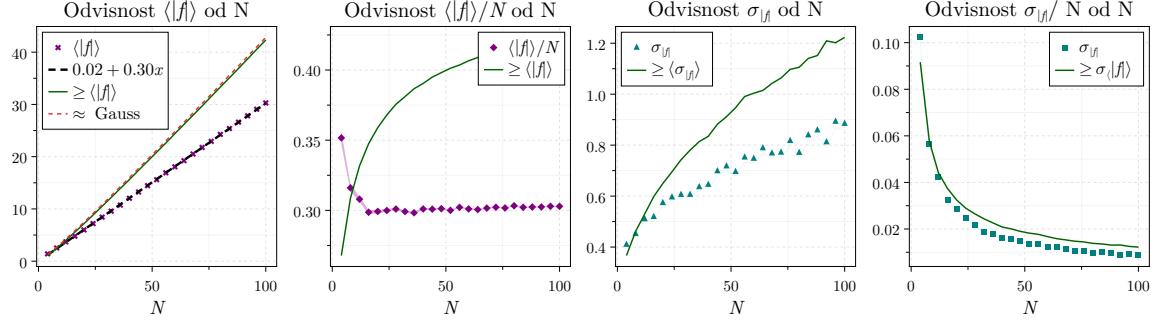
Verjetnost, da je vrednost manjša od minimuma $N - 1$ neodvisnih naključnih spremenljivk, je namreč enaka verjetnosti, da so vse te spremenljivke večje od dane vrednosti. Zaradi neodvisnosti to ustreza potenci posamezne verjetnosti. Pričakovano vrednost $|f|_{\max}$ dobimo iz definicije pričakovane vrednosti:

$$\mathbb{E}[|f|_{\max}] = \int_{-\infty}^{\infty} \mathcal{P}(|f|_{\max} > x) dx = \int_{-\infty}^{\infty} (\mathcal{P}(H > x))^{N-1} dx \quad (12)$$

Ta integral je težko izračunljiv saj je H_i porazdeljena kot N -ta konvolucija porazdelitve $c_{i,j}$. Za velike N bo po Centralnem Limitnem Izreku veljalo, da bo H_i v limiti $N \rightarrow \infty$ porazdeljen kot Gaussovka z $\mu = \sum_{i=1}^N \mathbb{E}[c_i]$ in $\sigma^2 = \sum \text{Var}_{i=1}^N [c_i]$. V primeru enakomerne porazdelitve je $\mu = \frac{N}{2}$ in $\sigma^2 = \frac{N}{12}$. Preko CLI ocenimo integral (12):

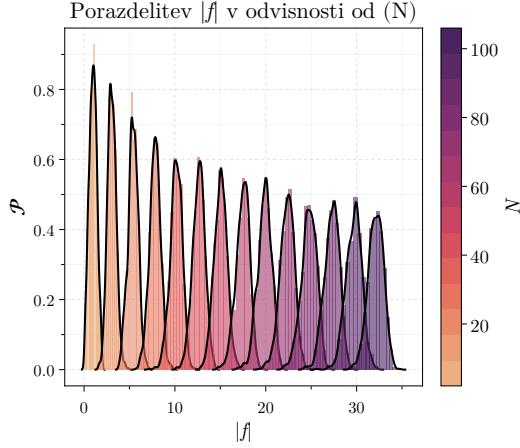
$$\mathbb{E}[|f|_{\min}] \approx \mu + \sigma \cdot \Phi^{-1}\left(\frac{1}{N}\right) \approx \mu_N + \sigma_N \left(-\sqrt{2 \log N} + \mathcal{O}\left(\frac{1}{\sqrt{\log N}}\right)\right) \quad (13)$$

Preko enačbe (13) smo dokazali, da je omrežje na vzgoraj omejeno z vrednostjo $\frac{|f|}{N} \leq \frac{1}{2}$, saj velja $|f| \leq \min \text{cut} \leq \min_i \{H_i\}$. Za več detajlov glej [ocena minimuma Gaussovka](#). Preverimo veljvanost naše izpeljavo na **Sliko 8**.



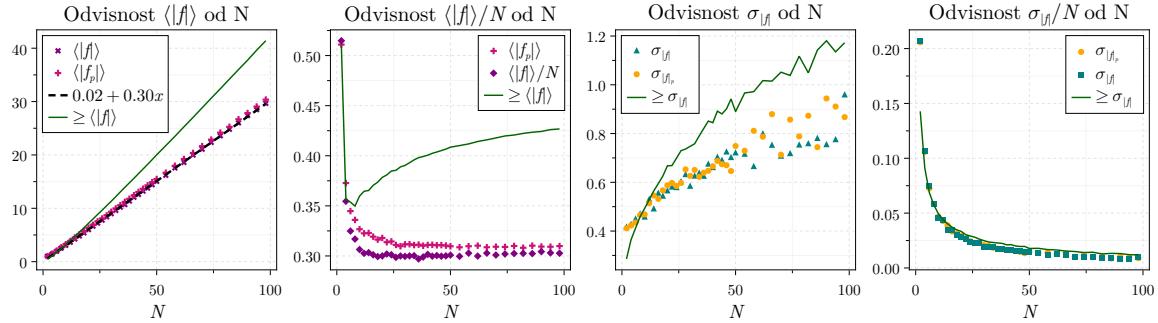
Slika 8: Rezultati, odvisnosti pretoka od velikosti N . Dodatno tudi prikažemo približel zgornje meje izpeljan v enačbi (13) in točno zgornjo mejo dobljeno s metodo Monte Carlo. Odstopanje napovedi povprečja za $N < 4$ (namesto $N - 1$ smo vstavili N).

Ker smo na **Sliki 6**, videli rast razmerja $\frac{|f|}{N}$ sumimo, da se z večanjem sistema približujemo omenjeni meji $\frac{|f|}{N} = 0.5$. V danem času nisem našel primerne spodnje meje, da bi to tudi dokazali.



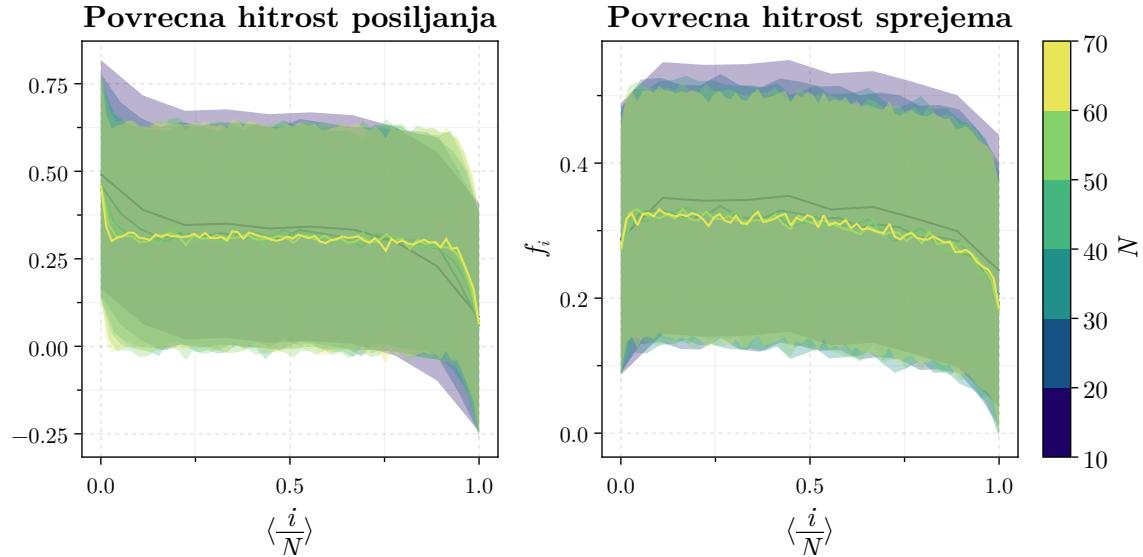
Slika 9: Porazdelitev maksimalnega pretoka v odvisnosti od velikosti sistema.

Vidimo tudi, da odvod variance pada z večanjem N . Te rezultate razumemo sledeče. Z večanjem sistema ima pretok večjo izbiro poti, zato so maksimumi pretoka manj razpršeni glede na manjša omrežja. Porazdelitev maksimalnega pretoka v odvisnosti od N prikazuje **Slika 9**. Porazdelitev se z večanjem N približuje Gaussovki. Zdela se mi je zanimivo poiškusiti **spremeniti topologijo** omrežja in primerjati rezultate. V trenutni obliki omrežja sta levi in desni predel oddaljena za vsaj N povezav. Zdi se, da bi bil pretok bolj učinkovit v primeru, da **periodično** povežemo robova. Rezultete kjer smo dodali **periodično** povezavo, prikazuje **Slika 10**.

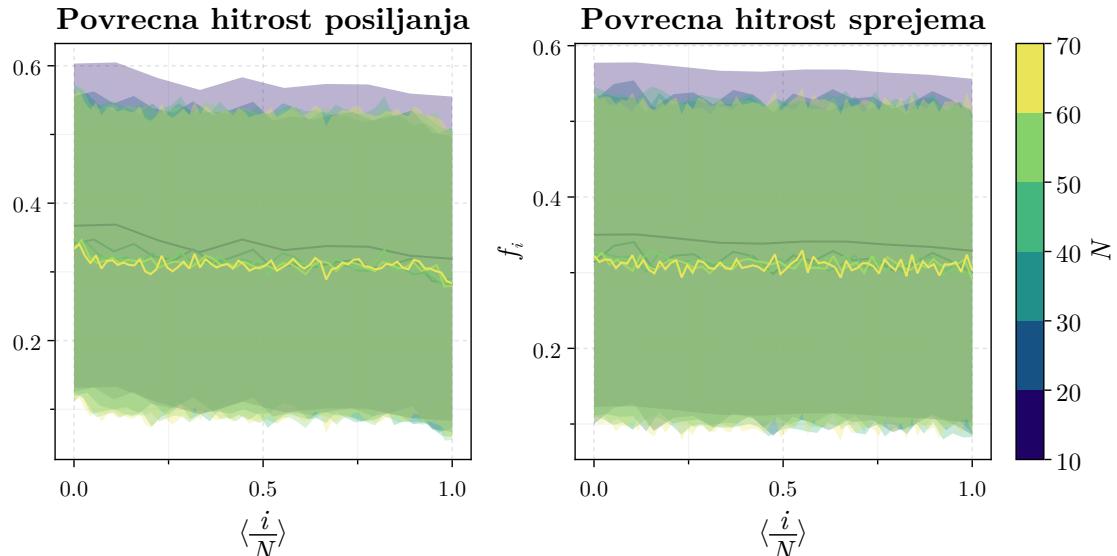


Slika 10: Pretok omrežja s periodično povezavo levega in desnega roba. Novi rezultati prikazani z roza in rumeno barvo ter označeni kot $|f|_p$ in $\sigma_{|f|_p}$, kažejo izboljšanje maksimalnega pretoka ob uvedbi periodične povezave. Pri manjših vrednostih N so izboljšave tudi do 10 %.

Vidimo, da periodična povezava izboljša pretok, vendar je izboljšava skoraj da zanemarljiva za večja omrežja. Analizirajmo sedaj še povprečne hitrosti odjemalcev in strežnikov, ko je dosežen maksimalni pretok. Na **Sliki 11** je prikazana povprečna hitrost posameznega odjemalca in strežnika v odvisnosti od mesta na mreži (od leve proti desni). Zanimivo, mi je bilo primerjati rezultate z algoritmom *Edmonds-Karp* na **Sliki 13**. Vidimo, da algoritmom *preflow* na **Sliki 11** pogosteje izbrere rešitev z velikostjo toka večjo na levi strani kot na desni, ko je navoljo več veljavnih rešitev. Pri algoritmu *Edmonds-Karp* je to manj opazno. Zdalo se mi je nekoliko nesmiselno poglobljeno analizirati distribucije hitrosti v primeru, da so te zgolj posledica algoritmov in ne lastnosti sistema. Rezultate sem primerjal med algoritmi in se osredotočil na univerzalne lastnosti sistema.

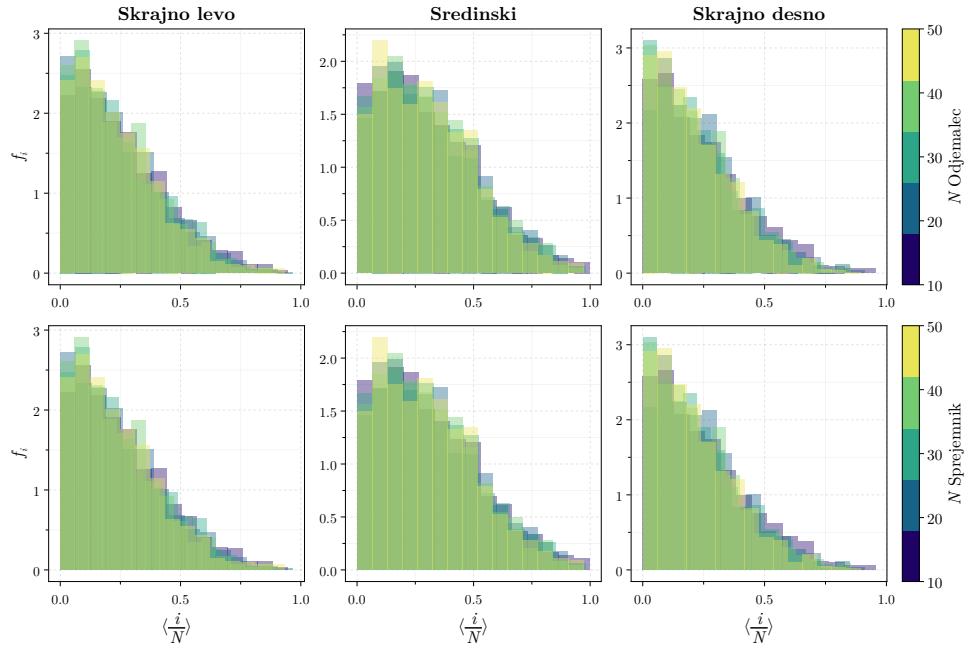


Slika 11: Prikaz povprečne hitrosti sprejemnikov in odjemalnikov v odvisnosti od indexa polovžaja. Z mejami

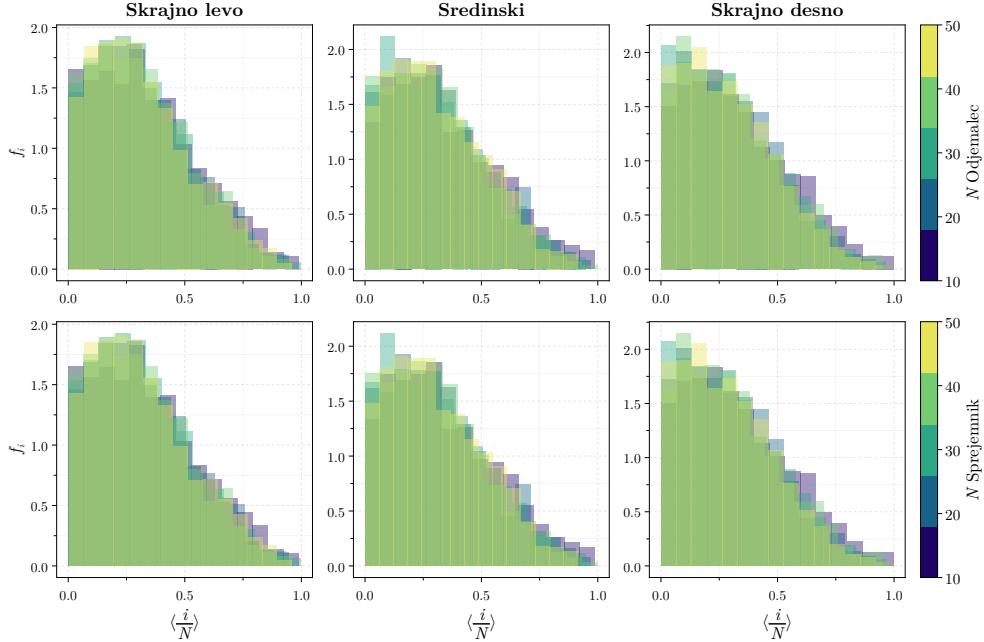


Slika 12: Prikaz povprečne hitrosti sprejem in oddajanja v odvisnosti od indexa polovžaja. Uporaba Edmonds-Karp algoritma.

Rezultate porazdelitev hitrosti za skrajno levi, sredinski in skrajno desni strežnik ter odjemalec prikazuje **Slika 13** (preflow), ki jo lahko primerjamo s **Sliko 14** (Edmonds-Karp).

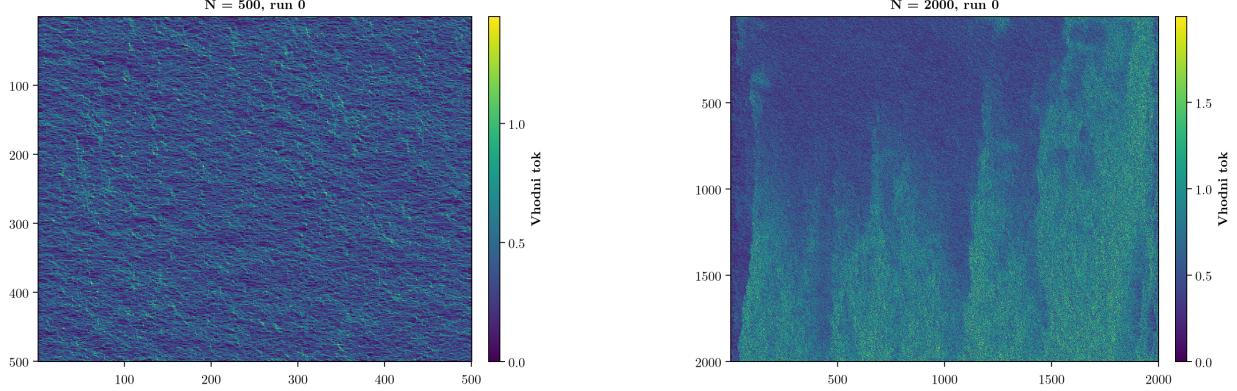


Slika 13: Porazdelitev hitrosti odjemalcev in strežnikov na skrajno desnem, sredinskem in skrajno levem položaju. Uporba preflow algoritma



Slika 14: Porazdelitev hitrosti odjemalcev in strežnikov na skrajno desnem, sredinskem in skrajno levem položaju. Uporba *Edmonds-karp* algoritma

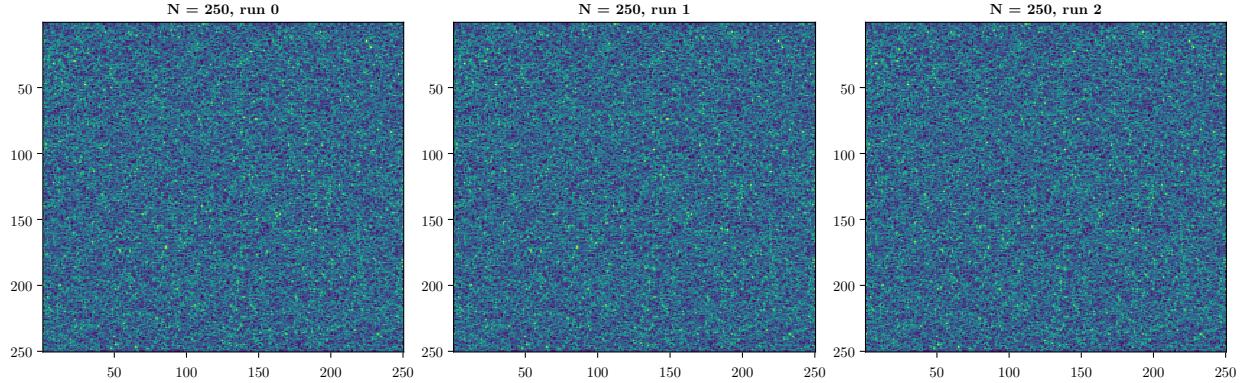
Porazdelitve delujejo neodvisne od velikosti omrežja, kar se mi je zdelo presenetljivo. Pri vseh histogramih smo imeli ansambel velikosti 1000. Preferenca algoritmov je razvidna tudi iz rešitev na **Sliki 15** in **Sliki 16**, kjer je prikazan heatmap vhodnih tokov v vsako vozlišče, to je skupna količina f , ki priteka v vozlišča.



Slika 15: Vhodni tok rešitev za eno izmed rešitev pri $N = 500$ z uporabo Edmonds-karp algoritma

Slika 16: Vhodni tok rešitev za eno izmed rešitev pri $N = 2000$ z uporabo preflow algoritma

Dodatno si oglejmo nekaj rešitev vhodnih tokov, ki smo jih rešili z linearnim programiranjem (**Slika 17**).

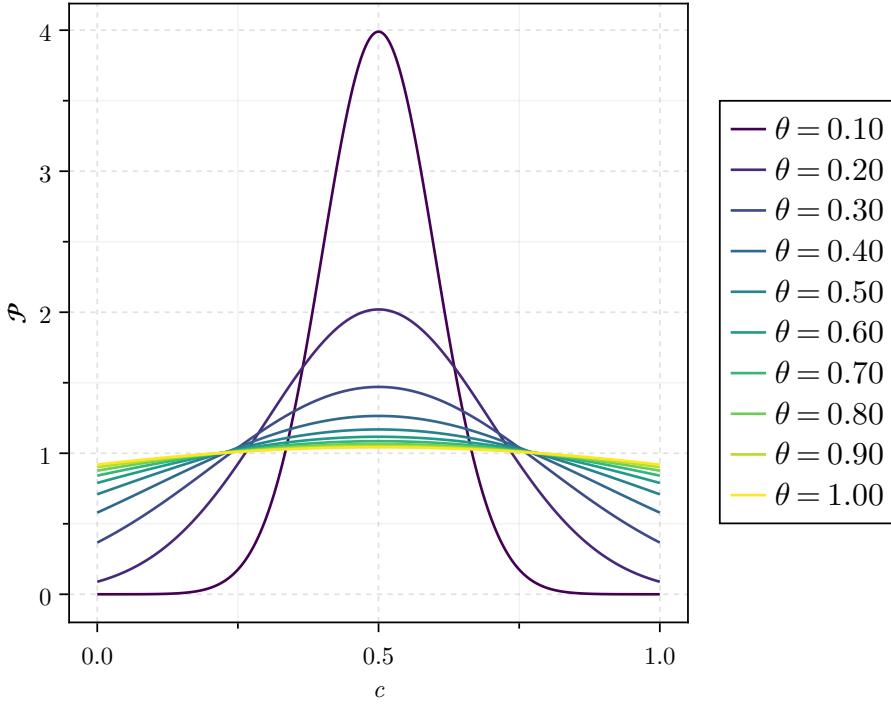


Slika 17: Razporeditev vhodnega toka za 3 neodvisne primere pri $N = 250$, rešeno s uporabo linearnega programiranja.

4.1 Različne distribucije

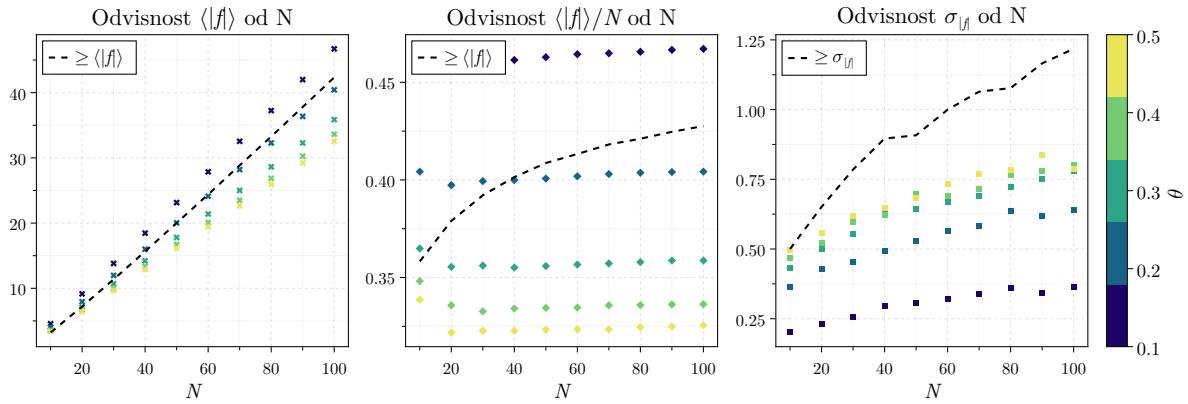
Ogljemo si kako izbira verjetnostne porazdelitve kapacitet vpliva na lastnosti omrežja. Iz enačbe (13) sledi, da je zgornja meja maksimalnega pretoka omejena s povprečno vrednostjo porazdelitve. Da bi bile primerjave smiselne smo izbral Gaussovo porazdelitev z $\mu = 0.5$ in spremojali varianco ter ohranjali porazdelitev na intervalu $[0,1]$. Primer porazdelitve, ki jo dobimo na opisan način prikazuje **Slika 18**. Parameter θ opisuje standarden odklone omejene Gaussovke na interval $[0,1]$.

Zožitev Gaussovke ($\mu = \frac{1}{2}$, $\sigma = \theta$) na interval $[0,1]$



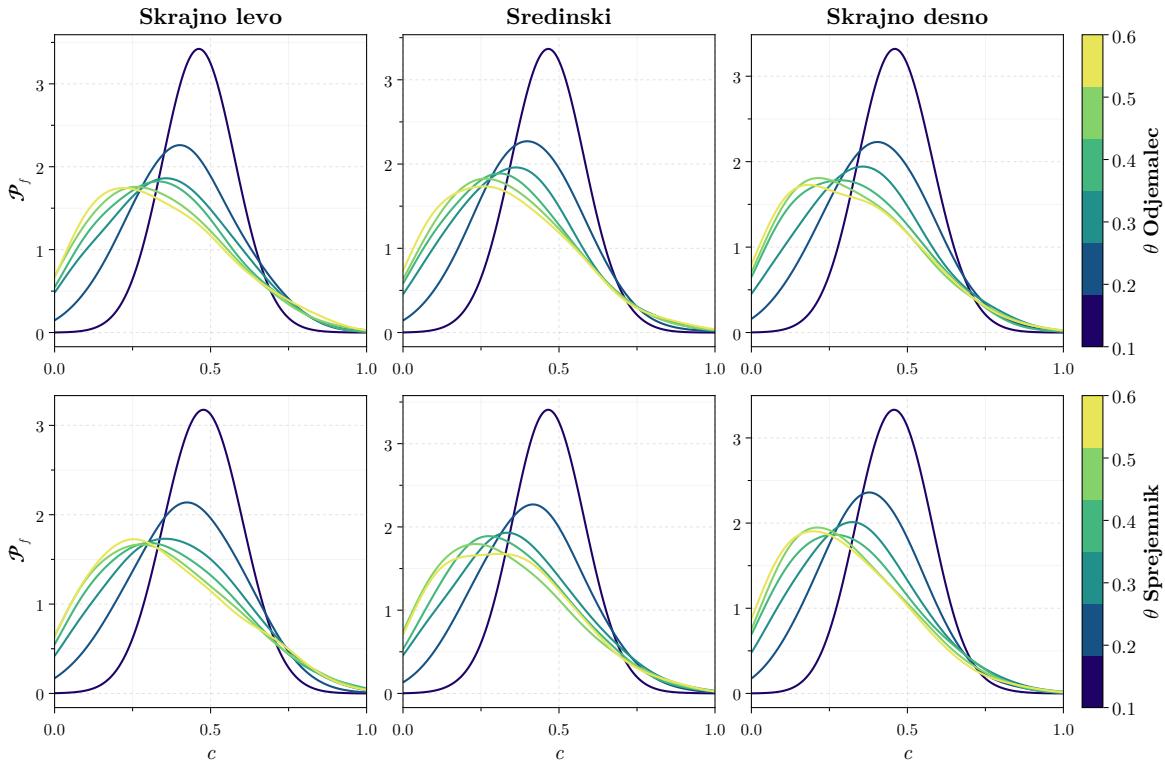
Slika 18: Primer porazdelitve, kjer ohranjamo povprečno vrednost in spremenjamo variacno a pri tem ohranjamo definicijsko območje na $[0,1]$.

Lastnosti maksimalnega pretoka v odvisnosti od θ prikazuje **Slika 19**. Opazimo, da se maksimalni pretok z zmanjšanjem variance izboljša in približa teoretični limiti $\frac{|f|}{N} \rightarrow \frac{1}{2}$. Rezultat je smiselen, saj je limitni primer to, da imajo vse povezave kapacite $c = 0.5$.



Slika 19: rezultati maksimalnega pretoka v odvisnosti od parmetra verjetnostne porazdelitve θ (**Slika 18**).

Slika 20 prikazuje porazdelitev odjemalcev in strežikov v odvisnosti od θ . Rezultati imajo za manjše θ porazdelitev Gaussovke, ki se z večanjem variance premika k robu.

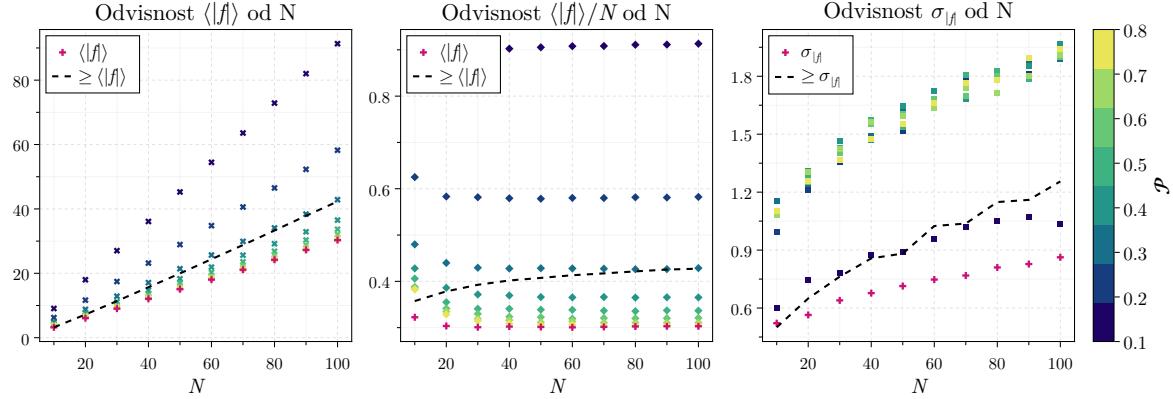


Slika 20: Porazdelitev hitrosti strežnikov in odjemalec v odvisnosti od parametra θ .

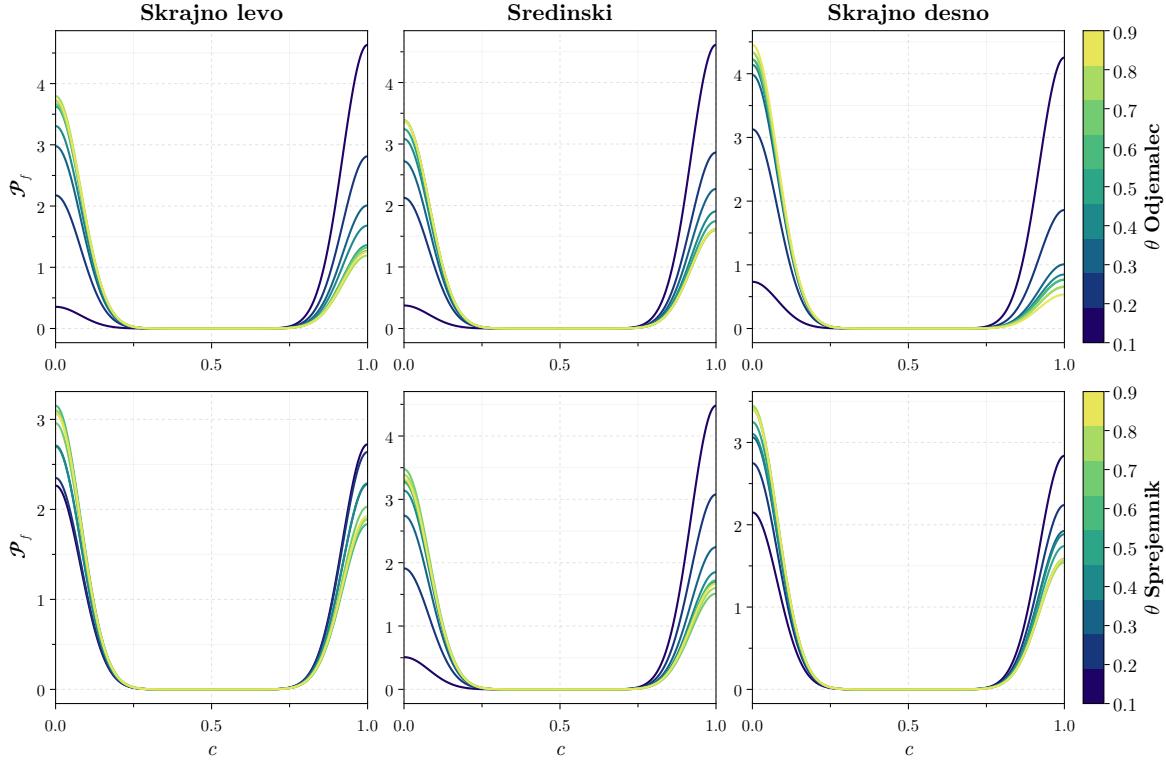
Rezultate bi lahko interpretirali na sledeč način. Recimo, da konstruiramo omrežje, katerega členi niso najbolj zanesljivi (naključno porazdeljena hitrost prenosta). Iz rezultatov je razvidno, da je omrežje stabilnejše in hitrejše ($\mu = 0.5$) v primeru, ko ima porazdelitev manjšo varianco. Tukaj se mi je zdelo zanimivo vprašati, kaj pa v primeru, ko imamo namesto naključnih povezav, stabilne povezave, ki pa z neko verjetnostjo obstajajo.

4.2 Naključen graf

Skonstruirajmo omrežje, ki ima povezave s konstantno kapaciteto $c = 1$, vendar povezava obstaja z verjetnostjo $1 - \mathcal{P}$. Rezultati te analize prikazujeta **Slika 21** in **Slika 22**. Vidimo, da je maksimalen pretoka večji od maksimalnega pretoka na grafu s naključnimi kapacetetami. Takšno omrežje se izkaže kot učinkovitejše pri doseganju višjega povprečnega pretoka, vendar bistveno slabše glede na varianco. Iz porazdelitev hitrosti na **Sliki 21** je razvidno, da je takšno omrežje z vidika posameznika manj zanesljivo, saj bi bil večino časa brez dostopa do omrežja.



Slika 21: Maksimalni pretok v odvisnosti od verjetnosti \mathcal{P} , da povezava med vozlišči ni prisotna. Rezultati za primer z naključnimi kapacitetami so prikazani z roza barvo.



Slika 22: Porazdelitev hitrosti omrežja za naključni graf, kjer posamezne povezave manjkajo z verjetnostjo \mathcal{P} . Rezultati kažejo, da je hitrost oddajanja običajno ali zelo visoka ali pa je sploh ni.

5 Zaključek

Obravnavali smo model omrežja z naključnimi povezavami. Ugotovili smo, da se maksimalni pretok poveča, če omrežju dodamo periodične povezave med levim in desnim robom ter izberemo poraz-

elitev z minimalno varianco. Izpeljali smo tudi teoretično mejo za velikost maksimalnega pretoka. Model smo primerjali z naključnim grafom s stabilnimi povezavami, ki obstajajo z določeno verjetnostjo. V tem primeru je sicer pretok večji, vendar je omrežje z vidika posameznega uporabnika manj zanesljivo. Model bi lahko dodatno razširili z uvedbo usmerjenega prenosa podatkov proti določenemu sprejemniku. Tako bi bolje modelirali realistične scenarije, v katerih je lokacija cilja ključna za oceno učinkovitosti omrežja.

Literatura

- [1] Brent Austgen. Max-flow/min-cut duality: Formulation & theory, 2023. URL <https://www.brentaustgen.com/blogs/maxflow-mincut-duality-1/>. Dostopano: 2025-08-10.
- [2] Jeff Bezanson, Alan Edelman, Stefan Karpinski, and Viral B Shah. Julia: A fresh approach to numerical computing. *SIAM Review*, 59(1):65–98, 2017. doi: 10.1137/141000671. URL <https://pubs.siam.org/doi/10.1137/141000671>.
- [3] Miles Lubin, Oscar Dowson, Joaquim Dias Garcia, Joey Huchette, Benoît Legat, and Juan Pablo Vielma. JuMP 1.0: Recent improvements to a modeling language for mathematical optimization. *Mathematical Programming Computation*, 15:581–589, 2023. doi: 10.1007/s12532-023-00239-3.
- [4] Q. Huangfu and J. A. J. Hall. Parallelizing the dual revised simplex method. *Mathematical Programming Computation*, 10(1):119–142, 2018. ISSN 1867-2957. doi: 10.1007/s12532-017-0130-5. URL <https://doi.org/10.1007/s12532-017-0130-5>. Published 2018-03-01.
- [5] Balázs Dezső, Alpár Jüttner, and Péter Kovács. Lemon – an open source c++ graph template library. *Electronic Notes in Theoretical Computer Science*, 264(5):23–45, 2011. ISSN 1571-0661. doi: <https://doi.org/10.1016/j.entcs.2011.06.003>. URL <https://www.sciencedirect.com/science/article/pii/S1571066111000740>. Proceedings of the Second Workshop on Generative Technologies (WGT) 2010.