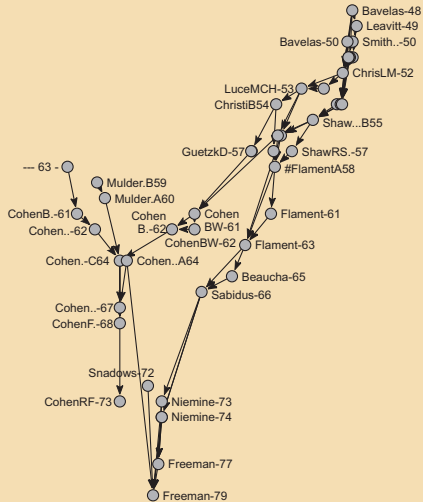


# Analiza omrežij citiranj



Andrej Mrvar

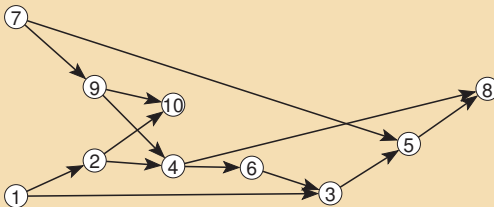


Poseben primer usmerjenega omrežja je **aciklično omrežje**:

Usmerjeno omrežje imenujemo *aciklično omrežje*, če v omrežju ni nobenega cikla.

Če začnemo pot v katerikoli točki acikličnega omrežja in sledimo smerem povezav, se v nobenem primeru ne moremo vrniti v začetno točko.

Primeri acikličnih omrežij: projektna omrežja, omrežja citiranj, rodovniki...

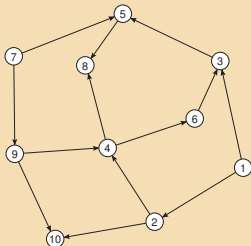


V acikličnem omrežju obstajajo **prve točke** – točke v katere ne vodi nobena povezava (točki 1 in 7 v zgornjem primeru) in **zadnje točke** – točke iz katerih ne vodi nobena povezava (točki 8 in 10 v zgornjem primeru).



Za vsako točko acikličnega omrežja lahko izračunamo **globino**: Začetnim točkam pripišemo globino 1, jih izločimo iz omrežja ter tako dobimo novo aciklično omrežje. Iz novodobljenega omrežja spet izločimo vse začetne točke, jim pripišemo globino 2 in postopek nadaljujemo.

**Primer:** Za točke v acikličnem omrežju



so **globine**:

vertex	1	2	3	4	5	6	7	8	9	10
depth	1	2	5	3	6	4	1	7	2	3



Pajek

# Aciklična omrežja...

...Uvod

Omrežja citiranj

Aciklična omrežja

Uteži citiranj

Algoritem

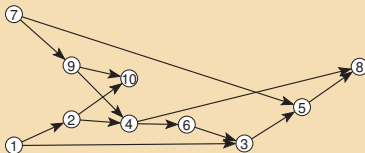
Različice

Primeri

Pajek

Naloge

Dobljene **globine** lahko uporabimo za **prikaz omrežja po nivojih** (v smeri  $x$ ):



V Pajku izračunamo globine točk acikličnega omrežja z

**Network / Acyclic Network / Depth Partition / Acyclic**

Če nato uporabimo ukaz **Draw / Network + First Partition**, se na sliki omrežja točke pobarvajo z barvo, ki ustreza globini. Uporabljene barve lahko pogledamo z:

**Options / Colors / Partition Colors / for Vertices.**

Sliko narišemo po nivojih z ukazom **Layers / in y Direction**. Če želimo sliko ročno popravljati, a bi pri tem radi ohranili nivojsko sliko, lahko proglasimo koordinato  $y$  za nepremično z ukazom **Move / Fix y** – s tem onemogočimo spremembo koordinate  $y$  katerekoli točke.

Nivoje v smeri  $x$  (od leve proti desni) dobimo z rotiranjem slike za 90 stopinj: **Options / Transform / Rotate 2D / 90**



Pajek

# Aciklična omrežja...

Omrežja citiranj...

Omrežja citiranj

Aciklična omrežja

Uteži citiranj

Algoritmi

Različice

Primeri

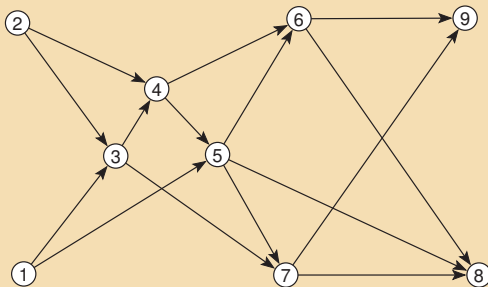
Pajek

Naloge

*Vhodni podatki:* Aciklično omrežje, ki predstavlja citiranje med članki, knjigami ali drugimi publikacijami.

*Rezultat:* Relativne pomembnosti člankov in relativne pomembnosti citiranj.

Primer omrežja citiranj:



V omrežju citiranj predstavljajo točke članke, usmerjena povezava od članka  $x$  do članka  $y$  pomeni, da  $y$  citira  $x$  ( $x$  je citiran v  $y$ ).



Pajek

# ...Aciklična omrežja...

...Omrežja citiranj...

Omrežja citiranj

Aciklična omrežja

Uteži citiranj

Algoritmi

Različice

Primeri

Pajek

Naloge

Ugotoviti želimo, kateri članek je najpomembnejši in katera povezava oz. citiranje je bistveno za celotno omrežje.

Najpreprostejša mera pomembnosti je, da preštejemo kolikokrat je dani članek citiran. V naši obliki predstavitve to pomeni, da preštejemo koliko puščic vodi iz dane točke.

Število citiranj (število puščic iz dane točke) dobimo z:

**Network / Create Partition / Degree / Output**

ali

**Network / Create Vector / Centrality / Degree / Output**

Glede na ta kriterij je najpomembnejši članek 5 s tremi citiranjmi.

---

Boljša mera za pomembnost točke v omrežju je zgrajena na osnovi števila poti, ki vodijo skozi dano točko.



Pajek

# ...Aciklična omrežja...

...Omrežja citiranj...

Omrežja citiranj

Aciklična omrežja

Uteži citiranj

Algoritem

Različice

Primeri

Pajek

Naloge

## Algoritem za štetje vseh poti v omrežju

- Omrežju dodamo enolični začetek in enolični konec.
- Opravimo dva prehoda skozi omrežje:
  - 1 V prvem prehodu gremo od začetne do končne točke in si za vsako točko zapomnimo, koliko poti vodi vanjo iz začetne točke (to ugotovimo z zaporednim prištevanjem poti glede na točke iz katerih pridemo v dano točko).
  - 2 V drugem prehodu gremo od končne točke nazaj proti začetni in si za vsako točko zapomnimo, koliko poti vodi iz te točke proti končni točki.



Pajek

# ...Aciklična omrežja...

...Omrežja citiranj...

Omrežja citiranj

Aciklična omrežja

Uteži citiranj

Algoritem

Različice

Primeri

Pajek

Naloge

Iz dobljenih podatkov lahko za vsako točko v omrežju ugotovimo število vseh poti od začetne do končne točke, ki vodijo skozi to točko. Uporabimo osnovni izrek kombinatorike:

*Če iz začetne do izbrane točke vodi  $M$  poti, od izbrane do končne pa  $N$  poti, potem je število vseh poti, od začetne do končne, ki gredo skozi izbrano točko, enako  $MN$ .*

Podobno ugotovimo koliko poti uporablja dano povezavo. Naj vodi izbrana povezava od točke  $i$  do točke  $j$ .

*Če iz začetne točke vodi v točko  $i$   $M$  različnih poti, iz točke  $j$  pa vodi v končno točko  $N$  različnih poti, potem je število poti, ki uporablja povezavo  $(i, j)$ , enako  $MN$ .*

Za lažje primerjanje rezultatov iz različnih omrežij ponavadi omenjena števila delimo s številom vseh poti v omrežju, tako da dobimo pomembnosti člankov/citiranj v intervalu  $[0, 1]$ .





Pajek

# ...Aciklična omrežja...

...Omrežja citiranj...

Omrežja citiranj

Aciklična omrežja

Uteži citiranj

Algoritem

Različice

Primeri

Pajek

Naloge

Obstajajo tri variante algoritma:

- metoda **Search Paths Count (SPC)** – to je ravnokar opisana metoda (štejemo poti od začetka do konca);
- metoda **Search Path Link Count (SPLC)** – ta metoda se od prejšnje loči po tem, da vsako točko v omrežju obravnavamo kot začetek, zato iz enoličnega začetka dodamo še povezave do vseh točk omrežja – število vseh poti v omrežju se poveča. Po tej metodi dobijo točke na koncu omrežja višje uteži.
- metoda **Search Path Node Pair (SPNP)** – vsako točko v omrežju obravnavamo kot začetek in kot konec, zato iz enoličnega začetka dodamo povezave do vseh točk omrežja in iz vsake točke omrežja dodamo še povezave do enoličnega konca. Zaradi tega dobijo točke in povezave v sredini omrežja višje uteži.



*Pajek*

# ...Aciklična omrežja...

...Omrežja citiranj...

Omrežja citiranj

Aciklična omrežja

Uteži citiranj

Algoritmi

Različice

Primeri

Pajek

Naloge

## Absolutne in relativne pomembnosti člankov po metodah SPC, SPLC in SPNP

čl.	<i>SPC</i>	<i>SPC/30</i>	<i>SPLC</i>	<i>SPLC/58</i>	<i>SPNP</i>	<i>SPNP/99</i>
1	14	0.47	14	0.24	25	0.25
2	16	0.53	16	0.28	29	0.29
3	18	0.60	27	0.47	48	0.48
4	21	0.70	35	0.60	60	0.61
5	20	0.67	35	0.60	56	0.57
6	14	0.47	26	0.45	39	0.39
7	12	0.40	22	0.38	33	0.33
8	17	0.57	32	0.55	32	0.32
9	13	0.43	25	0.43	25	0.25



Pajek

# ...Aciklična omrežja...

...Omrežja citiranj...

Omrežja citiranj

## Absolutne in relativne pomembnosti citiranj po metodah SPC, SPLC in SPNP

Aciklična omrežja

Uteži citiranj

Algoritem

Različice

Primeri

Pajek

Naloge

od	do	SPC	SPC/30	SPLC	SPLC/58	SPNP	SPNP/99
1	3	9	0.30	9	0.16	16	0.16
2	3	9	0.30	9	0.16	16	0.16
1	5	5	0.17	5	0.09	8	0.08
2	4	7	0.23	7	0.12	12	0.12
3	7	4	0.13	6	0.10	9	0.09
3	4	14	0.47	21	0.36	36	0.36
4	5	15	0.50	25	0.43	40	0.40
4	6	6	0.20	10	0.17	15	0.15
5	7	8	0.27	14	0.24	21	0.21
5	6	8	0.27	14	0.24	21	0.21
5	8	4	0.13	7	0.12	7	0.07
6	8	7	0.23	13	0.22	13	0.13
6	9	7	0.23	13	0.22	13	0.13
7	8	6	0.20	11	0.19	11	0.11
7	9	6	0.20	11	0.19	11	0.11



Pajek

# ...Aciklična omrežja...

...Omrežja citiranj

Omrežja citiranj

Aciklična omrežja

Uteži citiranj

Algoritmi

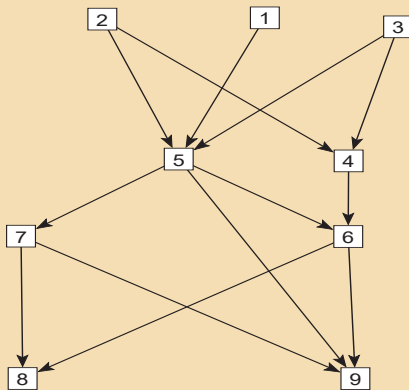
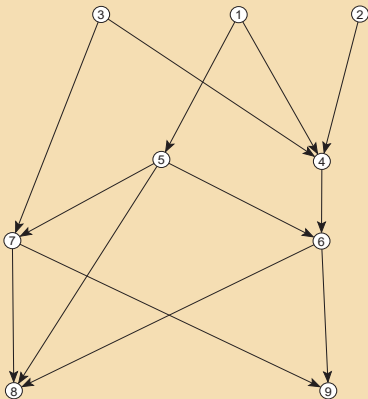
Različice

Primeri

Pajek

Naloge

## Še dva manjša primera: citem.net in Citem2a.net





Pajek

# ...Aciklična omrežja...

Omrežja citiranj v Pajku...

Omrežja citiranj

Pomembnosti člankov in citiranj izračunamo z ukazom:

**Network / Acyclic Network / Create Weighted Network + Vector / Traversal Weights / SPC ali SPLC ali SPNP**

Aciklična omrežja

Uteži citiranj

Algoritem

Različice

Primeri

Pajek

Naloge

Pajek vrne naslednje rezultate:

- 1 Vektor s številom poti, ki vodijo iz začetne točke v vsako točko (*Number of Different Incoming Paths*).
- 2 Vektor s številom poti, ki vodijo iz vsake točke v končno točko (*Number of Different Outgoing Paths*).
- 3 Vektor s pomembnostmi člankov.
- 4 Omrežje, kjer vrednosti na povezavah predstavljajo pomembnosti citiranj (*Citation weights*). Rezultate pogledamo v oknu **Draw**, kjer povezave označimo z vrednostmi: **Options / Lines / Mark Lines / with Values**. Če želimo ugotoviti, katera citiranja so najpomembnejša, poženemo **Network / Info / General** in odtipkamo koliko najvišjih vrednosti želimo videti (npr. če želimo pogledati 10 najvišjih, vnesemo 10).



Pajek

# ...Aciklična omrežja...

...Omrežja citiranj v Pajku...

Omrežja citiranj

Aciklična omrežja

Uteži citiranj

Algoritmi

Različice

Primeri

Pajek

Naloge

## Določanje glavne poti od začetne do končne točke

Izračunane uteži pa ne uporabimo samo za določanje najpomembnejših člankov in citiranj, ampak tudi za iskanje **glavne poti**. Glavna pot je pot od začetne do končne točke, ki vsebuje največje uteži citiranj. Obstaja več metod za določanje glavne poti. Delimo jih na **lokalne** in **globalne**.

**Lokalna iskanja** – Iščemo korak za korakom: v vsakem koraku za doseženo točko izberemo povezavo, ki vodi iz te točke in ima največjo utež.

**Globalna iskanja** – Iščemo pot od začetne do končne točke z največjo skupno vsoto uteži.

**Network / Acyclic Network / Create (Sub)Network / Main Paths**



**Lokalne glavne poti:** v vsakem koraku iskanja izberemo povezavo (povezave) z največjo utežjo, ki so sosednje trenutni povezavi. Dve varianti lokalnega iskanja:

- 1 Local / Forward** – poiščemo lokalne glavne poti od začetnih do končnih točk.
- 2 Local / Backward** – poiščemo lokalne glavne poti v nasprotni smeri (od končnih do začetnih točk).

Lahko uporabimo tudi toleranco (v primeru normalizacije *flow* je to število med 0 in 1). Toleranca 0 pomeni, da v vsakem koraku upoštevamo samo povezave z največjo utežjo, medtem ko večje tolerance pomenijo, da upoštevamo tudi povezave z malenkost manjšimi utežmi.

Primer: Če za toleranco izberemo število 0.1 in je največja vrednost na povezavi 0.5, potem pridejo v poštev vse povezave z vrednostjo večjo od 0.45 (10% manjše vrednosti od 0.5).



*Pajek*

# ...Aciklična omrežja...

...Omrežja citiranj v Pajku

Omrežja citiranj

Aciklična omrežja

Uteži citiranj

Algoritmi

Različice

Primeri

Pajek

Naloge

**Globalne glavne poti:** poiščemo glavne poti z največjo skupno vsoto povezav na poti.

- 1 **Global / Standard** – poiščemo glavne poti od začetnih do končnih točk z največjo skupno vsoto povezav na poti. Ta postopek se imenuje tudi Metoda kritične poti.





Pajek

# ...Aciklična omrežja...

...Omrežja citiranj v Pajku

Omrežja citiranj

Aciklična omrežja

Uteži citiranj

Algoritem

Različice

Primeri

Pajek

Naloge

## Key Routes

Iščemo lahko tudi lokalne in globalne glavne poti, ki vsebujejo izbrane citate (ne pozabimo: citati so predstavljeni s povezavami). Izbrani citati so ponavadi tisti z najvišjimi utežmi citiranj - imenujemo jih **key-routes**.

Glavne poti, ki vsebujejo 5 najpomembnejših citiranj poiščemo tako, da pri vprašanju, koliko najpomembnejših citiranj želimo vključiti v glavno pot, vnesemo 1-5

---

Namesto iskanja glavnih poti, ki vsebujejo izbrane povezave (citiranja), lahko iščemo tudi glavne poti, ki vsebujejo izbrane točke (članke).

V tem primeru najprej izberemo številke člankov, ki jih želimo vključiti v Cluster, nato poštenemo ukaz: **Through Vertices in Cluster**. Tako poiščemo glavne poti, ki vodijo skozi točke, ki se nahajajo v izbrani skupini (Cluster).



Analizirajte omrežje citiranj med članki na temo središčnosti v omrežjih ([centrality.literature.net](http://centrality.literature.net)). Leta 1979 je Linton Freeman objavil članek, ki opisuje številne mere središčnosti. Članek je postal eden od najpomembnejših člankov na tem področju. Freeman pa ni bil prvi, ki je napisal članek o središčnosti v omrežjih. Njegov članek je del razprave, ki sega nazaj do leta 1940. Usmerjene povezave, ki predstavljajo citiranja, kažejo od članka, ki je citiran, na članek, ki ga citira. Nekaj nasvetov:

- 1 Uporabite normalizacijo *flow*.
- 2 Preverite, kaj se zgodi, če namesto tolerance 0 uporabite 0.07.
- 3 Kot *key-routes* uporabite najprej samo najvišje citiranje, potem pa še najvišji dve citiranj.
- 4 Poiščite lokalne in globalne glavne poti, ki vsebujejo članka Leavitt (1949) in GilchSW (1954). To sta članka s številka 2 in 21.



# ...Aciklična omrežja...

## Naloga 1 - nekaj rezultatov

Omrežja citiranih

Aciklična omrežja

Uteži citiranj

Algoritmi

Različice

Primeri

Pajek

Naloge



Zgoraj: *forward local in key-route local* (samo en key-route).

Middle: *standard global in key-route global* (samo en key-route).

Bottom: *backward local*.



Pajek

# ...Aciklična omrežja

## Naloga 2

Omrežja citiranj

Aciklična omrežja

Uteži citiranj

Algoritem

Različice

Primeri

Pajek

Naloge

Omrežje citiranj lahko predstavlja tudi mentorstva: relacija med študentom in njegovim mentorjem. Omrežje `PhD.net` vsebuje povezave med doktorski študenti in njihovimi mentorji na področju teoretičnega računalništva; usmerjene povezave kažejo od mentorja do študenta. V razbitju `PhD_year.clu` so shranjene letnice, ko so bile disertacije zaključene. V omrežju poiščite glavne poti po vseh kriterijih.