Analiza velikih omrežij

Vladimir Batagelj
Univerza v Ljubljani, FMF, matematika

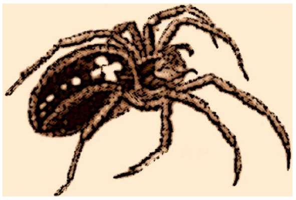
88. Solomonov seminar

IJS, 10. september 2002 ob 13h, Velika predavalnica

Spored

- Pajek
- Omrežja sklicevanj
- Omrežja sporočil

Pajek



`Pajek` je program, za Windows (32 bit), za analizo in prikaz *velikih omrežij*.

Pajek je prosto dostopen, za neposlovne uporabe, na naslovu:
`http://vlado.fmf.uni-lj.si/pub/networks/pajek/`

Program **`Pajek`** sva z Andrejem Mrvarjem s FDV začela razvijati novembra 1996. Nekaj postopkov je prispeval tudi Matjaž Zaveršnik.

Velika omrežja

Velika omrežja so tu!

- prometna omrežja in omrežja izmenjave sporočil;
- diagrami poteka programov;
- velike molekule;
- rodovniki;
- omrežja ustvarjena iz slovarjev in drugih besedil;
- bibliografije, omrežja sklicevanj, . . .

Ta omrežja imajo po več (sto) tisoč točk. Za njihovo analizo je bilo potrebno razviti nove, učinkovite postopke.

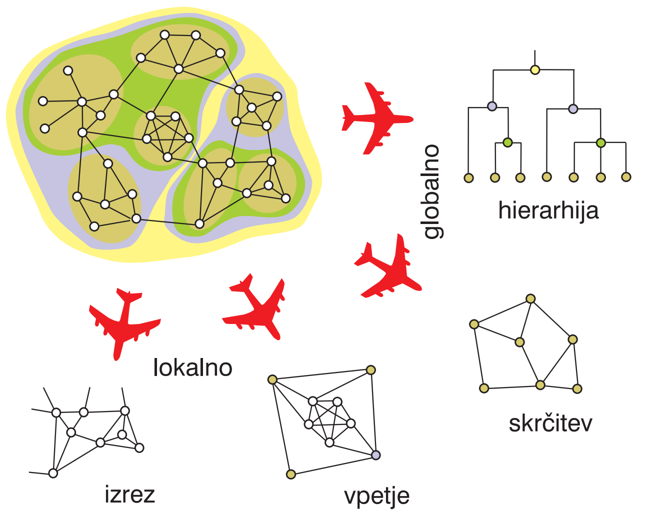
Pajek – Cilji

Glavni cilji pri izgradnji programa **Pajek** so:

- podpora *abstrakcije* – z (rekurzivnim) razčlenjevanjem velikega omrežja v večje število manjših omrežij, ki jih lahko nato obdelamo z zahtevnejšimi postopki;
- ponuditi uporabniku nabor zmogljivih orodij za pripravo *prikazov* omrežij;
- vgraditi izbor *učinkovitih* (podkvadratičnih) postopkov za analizo velikih omrežij; nekaj od njih smo razvili sami.

S programom **Pajek** lahko: *določimo* skupine (komponente povezanosti, okolice 'pomembnih' točk, sredice, . . .) v omrežju; *izrežemo* in ločeno *prikažemo* podomrežje porojeno z izbrano skupino, lahko tudi z vpetjem v omrežje; *skrčimo* točke iste skupine v eno točko in prikažemo povezave med skupinami.

Pajek – Cilji . . .



Podatkovne strukture programa Pajek

Trenutno program **Pajek** temelji na šestih vrstah podatkov:

- *omrežje* – osnovni podatki (točke in povezave);
- *razbitje* – kateri skupini pripada posamezna točka; imenski podatki o točkah;
- *vektor* – številski podatki o točkah;
- *permutacija* – urejenostni podatki o točkah;
- *skupina* – podmnožica množice točk;
- *hierarhija* – hierarhična urejenost na skupinah in točkah.

Moč programa **Pajek** temelji na večjem številu operacij, ki omogočajo različne pretvorbe med temi vrstami podatkov. Osnovno okno programa pajek je pravzaprav računalo nad njimi. Na njih temelji tudi razmestitev operacij po izbirah.

Pajek, poleg lastnih, podpira še več drugih oblik zapisa vhodnih podatkov: `UCINET DL`, rodovniški `GED`, opisi molekul: `BS` (Ball and Stick), `MAC` (Mac Molecule) in `MOL` (MDL MOLfile).

Primer

```
Read network DNA.BS
Net / Partitions / Vertex Labels
Draw / Draw-Partition
Options / Transform / Fit Area / max ...
Read network C5.net
Draw / Draw
Nets / First Network
[select] DNA network
Nets / Second Network
Nets / Fragment (1 in 2) / Find
Draw / Draw
```

Omrežja sklicevanj

Garfield s sodelavci je leta 1964 objavil članek, v katerem je na primeru omrežja sklicevanj zgrajenega na osnovi knjige Asimova o zgodovini DNA pokazal, da obstaja velika povezanost med vsebinsko analizo pomembnosti posameznih dogodkov in pomembnostjo, ki jo nakazuje njihova vmeščenost v omrežje.

Leta 1989 sta Hummon in Doreian predlagala tri mere (NPPC, SPLC, SPNP) – uteži na povezavah, ki naj bi omogočile postopkovno razpoznavo pomembnih delov (glavne poti) v omrežjih sklicevanj.

Na obisku pri Doreianu v Pittsburghu leta 1990/91 sem se s temi merami seznanil in za mero SPLC našel zelo učinkovit (linearen glede na število povezav) postopek izračuna.

Značilnosti omrežij sklicevanj

Na dani množici enot $\mathbf{U}$ (članki, knjige, dela, ...) vpeljemo relacijo *sklicevanja* $R \subseteq \mathbf{U} \times \mathbf{U}$: $uRv \equiv v$ se sklicuje na u, ki določa *omrežje sklicevanj* $\mathbf{N} = (\mathbf{U}, R)$.

V tabeli so zbrane osnovne značilnosti izbranih omrežij sklicevanj iz Garfieldove zbirke ustvarjene iz Web of Science s programom ***HistCite***.

omrežje	n	m	m_0	n_0	n_C	k_C	h	Δ_i	Δ_o	2	3	4
DNA	40	60	0	1	35	3	11	7	5	0	0	0
small world	396	1988	0	163	233	1	16	60	294	0	0	0
Small and Griffith	1059	4922	1	35	1024	1	28	89	232	2	0	0
Scientometrics	3084	10416	1	355	2678	21	32	121	105	5	2	1
self-organizing maps	4470	12731	2	698	3704	27	24	51	735	11	0	0
Zewail	6752	54253	1	101	6640	5	75	166	227	38	1	2
Lederberg	8843	41609	7	519	8212	35	63	135	1098	54	4	0

Nekaj pojmov o relacijah

Relacija sklicevanja je *irrefleksivna*, $\forall u \in \mathbf{U} : \neg uRu$, in (skoraj) *aciklična*. V nadaljnjem bomo privzeli to lastnost.

Naj bo $I = \{(u,u) : u \in \mathbf{U}\}$ *enotska* relacija na $\mathbf{U}$ in $\overline{R} = \bigcup_{k\in\mathbb{N}^+} R^k$ *tranzitivna ovojnica* relacije R. Tedaj je R aciklična ntk. $\overline{R} \cap I = \emptyset$.

Relacija $R^\star = \overline{R} \cup I$ je *tranzitivna in refleksivna ovojnica* relacije R.

Za relacijo $Q \subseteq \mathbf{U} \times \mathbf{U}$ označimo z Q^{inv} njej *obratno* relacijo, $uQ^{\mathrm{inv}}v \equiv vQu$, in z

$$Q(u) = \{v \in \mathbf{U} : uQv\}$$

množico *naslednikov* enote $u \in \mathbf{U}$. Če je Q is aciklična, je taka tudi Q^{inv}.

Lastnosti omrežij sklicevanj

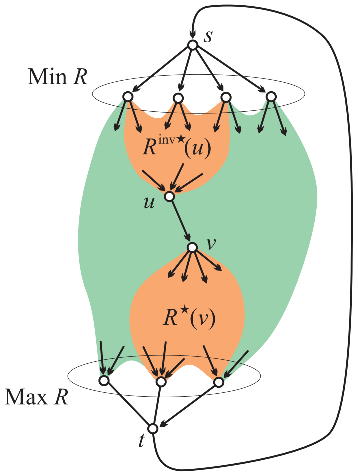
Ker je množica enot $\mathbf{U}$ končna in relacija R aciklična, velja:

- Množico enot $\mathbf{U}$ lahko *topološko uredimo* – obstaja surjektivna preslikava (permutacija) $i : \mathbf{U} \to 1..|\mathbf{U}|$, za katero velja

  $$uRv \Rightarrow i(u) < i(v)$$

- Naj bo $\mathrm{Min}\, R = \{u \in \mathbf{U} : R^{\mathrm{inv}}(u) = \emptyset\}$ množica *minimalnih* elementov in $\mathrm{Max}\, R = \{u \in \mathbf{U} : R(u) = \emptyset\}$ množica *maksimalnih* elementov. Tedaj $\mathrm{Min}\, R \neq \emptyset$ in $\mathrm{Max}\, R \neq \emptyset$.
- vsaka točka $u \in \mathbf{U}$ in vsaka povezava $(u,v) \in R$ ležita vsaj na eni poti iz $\mathrm{Min}\, R$ v $\mathrm{Max}\, R$:
  $\forall u \in \mathbf{U} : R^\star(u) \cap \mathrm{Max}\, R \neq \emptyset$
  $\forall u \in \mathbf{U} : R^{\mathrm{inv}\star}(u) \cap \mathrm{Min}\, R \neq \emptyset$

Izbrana oblika omrežij sklicevanj



Teorija in postopki se precej poenostavijo, če dano omrežje sklicevanj $(\mathbf{U}, R)$ dopolnimo do *izbrane oblike* $(\mathbf{U}', R')$, tako da množici enot dodamo skupni *izvor* s in skupni *ponor* t, $s, t \notin \mathbf{U}$, $\mathbf{U}' := \mathbf{U} \cup \{s, t\}$; ter ju ustrezno povežemo

$$R' := R \cup \{s\} \times \operatorname{Min} R \cup \operatorname{Max} R \times \{t\} \cup \{(t, s)\}$$

Hummon in Doreian-ove uteži

Posamezni povezavi priredimo utež $w : R \to \mathbb{R}_0^+$, ki jo uporabimo za določitev pomembnih delov omrežja.

- *node pair projection count* (NPPC): $w_1(u, v) = |R^{\text{inv}\star}(u)| \cdot |R^\star(v)|$;
- *search path count* (SPC): $w_2(u, v) = N(u, v)$, kjer je $N(u, v)$ število različnih poti iz $\operatorname{Min} R$ v $\operatorname{Max} R$ (ali iz s v t), ki vodijo skozi povezavo (u, v);
- *search path link count* (SPLC): $w_3(u, v) = N'(u, v)$, kjer je $N'(u, v)$ enak $N(u, v)$ za omrežje $(\mathbf{U}, R')$, $R' := R \cup \{s\} \times (\mathbf{U} \setminus \{s, t\})$;
- *search path node pair* (SPNP) method: $w_4(u, v) = L(u, v)$, kjer $L(u, v)$ šteje vse povezane pare po poteh, ki vodijo skozi povezavo $(u, v) \in R$.

Za NPPC lahko ustvarimo postopek zahtevnosti $O(nm)$, ki temelji na uporabi pregledovanja v širino za določitev $|R^{\text{inv}\star}(u)|$ in $|R^\star(v)|$.

Postopek za uteži SPC in SPLC

Za izračun $N(u, v)$ vpeljemo: $N^-(v)$ – število različnih poti iz $\operatorname{Min} R$ (ali s) do v, in $N^+(v)$ – število različnih poti iz v do $\operatorname{Max} R$ (ali t).

Po osnovnih pravilih kombinatorike je

$$N(u, v) = N^-(u) \cdot N^+(v), \qquad (u, v) \in R$$

in

$$N^-(u) = \begin{cases} 1 & u = s \\ \sum_{v : vRu} N^-(v) & \text{sicer} \end{cases}$$

and

$$N^+(u) = \begin{cases} 1 & u = t \\ \sum_{v : uRv} N^+(v) & \text{sicer} \end{cases}$$

Če računamo količini N^- in N^+ v vrstnem redu, ki ga določa (obratna) topološka urejenost, ju lahko izračunamo v času $O(m)$. Topološka urejenost zagotavlja, da se med računanjem vse količine, ki jih potrebujemo že znane (izračunane).

Postopek za uteži SPNP

Za uteži SPNP obstaja podoben postopek. Naj bo $c(\pi)$ število točk na poti π,

$$L^-(u) = \sum_{\substack{\pi \text{ is a path} \\ \text{from } \operatorname{Min} R \text{ to } u}} c(\pi) \quad \text{and} \quad L^+(u) = \sum_{\substack{\pi \text{ is a path} \\ \text{from } u \text{ to } \operatorname{Max} R}} c(\pi)$$

Tedaj za uRv velja

$$w_4(u, v) = L^-(u) \cdot L^+(v)$$

Postopek določitve $L^-(u)$ in $L^+(u)$ lahko pregledno opišemo z uporabo dveh družin polinomskih rodovnih funkcij $P^-(u; x) = \sum_{k=1}^{h(u)} p^-(u, k) x^k$ in $P^+(u; x) = \sum_{k=1}^{h^-(u)} p^+(u, k) x^k$, za $u \in \mathbf{U}$. $h(u)$ je globina točke u v omrežju $(\mathbf{U}', R')$, in $h^-(u)$ globina točke u v omrežju $(\mathbf{U}', R^{\text{inv}\prime})$. Koeficient $p^-(u, k)$ šteje poti dolžine k iz s do u, $p^+(u, k)$ pa poti dolžine k iz u do t.

Postopek za uteži SPNP …

Ponovno po osnovnih pravilih kombinatorike

$$P^-(u;x) = \begin{cases} 1 & u = s \\ x \cdot \sum_{v:vRu} P^-(v;x) & \text{sicer} \end{cases}$$

in

$$P^+(u;x) = \begin{cases} 1 & u = t \\ x \cdot \sum_{v:uRv} P^+(v;x) & \text{sicer} \end{cases}$$

Obe družini lahko določimo po teh zvezah v topološkem vrstnem redu v času največ $O(hm)$. Od tu pa:

$$L^-(u) = \sum_k k \cdot p^-(u,k) \qquad \text{and} \qquad L^+(v) = \sum_k k \cdot p^+(v,k)$$

V prvi programski izvedbi so pri večjih omrežjih števila `LongInt` ušla iz obsega. V novi izvedbi uporabljamo Delphijeva števila `Extended`. Zanje je mogoče pokazati, da so zadostna za omrežja velikosti $h \leq 1000$ in $\Delta_i \leq 10000$.

Lastnosti uteži

Pokazati je mogoče celo vrsto zanimivih lastnosti. Na primer:

Bodita $(\mathbf{U}, R_1)$ in $(\mathbf{U}, R_2)$ omrežji sklicevanj nad isto množico enot $\mathbf{U}$ in $R_1 \subseteq R_2$, tedaj

$$\begin{aligned} w_1(u,v;R_1) &\leq w_1(u,v;R_2) \\ N(u,v;R_1) &\leq N(u,v;R_2) \\ w_4(u,v;R_1) &\leq w_4(u,v;R_2) \end{aligned}$$

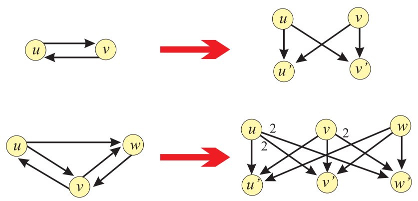
Za uteži $N(u,v)$ velja *Kirchoffov točkovni zakon*: Za vsako točko v omrežja sklicevanj (E,R) v izbrani obliki velja

$$\text{vhodni tok}(v) = \text{izhodni tok}(v)$$

Neaciklična omrežja sklicevanj

Običajno jih, ker so skoraj aciklična, predelamo v aciklična.

- skrčitev cikličnih skupin (netrivialne krepke komponente);
- odstranimo izbrane povezave;
- uporabo pretvorb kot je 'preprint'ska, ki temelji na: vsaka enota v krepki komponenti je podvojena s 'preprint'sko izdajo. Enote iz krepke komponente se sklicujejo na 'preprint'e.



Velike krepke komponente v omrežjih sklicevanj praviloma pomenijo napako v podatkih.

Primer: SOM (self-organizing maps)

```
File/Pajek Project File/Read [Kohonen.paj]
Net/Components/Strong [2]
Operations/Shrink Network/Partition [1][0]
Net/Transform/Remove/loops [yes]
Net/Citation Weights/Source-Sink
Macro/Play [ /macro/genea/layers.mcr ]
Draw/Draw-Partition { Main path }
Export/EPS/PS [path.eps]
Main: [select network: Citation weights]
Info/Network/Line Values [#100][No]
Net/Transform/Remove/lines with value/lower than [0.001][Yes]
Net/Partitions/Degree/All
Operations/Extract from Network/Partition [1][9999]
Vector/Extract Subvector [1][9999]
Macro/Play [ /macro/genea/layers.mcr ]
Draw/Draw-Partition-Vector
Layers/Resolution/High
Layers/Optimize Layers in x direction/Complete [yes][OK][OK]
Options/Mark Vertices Using/No Labels
Options/Lines/Different Widths
Options/Size/of Lines [25]
Options/Size/of Vertices [10]
Main: Net/Components/Weak [1]
Info/Partition [1][0]  {1 is the large component}
Operations/Extract from Network/Partition [1][1]
Vector/Extract Subvector [1][1]
Net/Partitions/Depth/Genealogical
Draw/Draw-Partition-Vector
Move/Fix/y
```

Primer: SOM (self-organizing maps) ...

```
Info/Network  {threshold 0.05 selected}
Net/Transform/Remove/lines with value/lower than [0.05]
Net/Partitions/Degree/All
Partition/Binarize [1][999]
[select the reduced network]
Draw/Draw-Partition-Vector
Options/Mark Vertices Using/Mark Cluster Only
Options/Mark Vertices Using/Labels
Export/SVG/Line Values/Options/GreyScale
Export/SVG/Line Values/Options/Different Widths
Export/SVG/Line Values/Nested Classes  [main.htm][#20]
Main: [select network: shrunk KohonenL.net]
Net/Vector/Important Vertices/1-Mode: Hubs&Authorities [15][15]
Info/Vector [15][Cancel]
[select vector: Hub Weights]
Info/Vector [15][Cancel]
```


Omrežje Reuters terror news

Steve Corman in Kevin Dooley z Arizona State University sta razvila nov postopek za analizo besedil imenovan ***Centering Resonance Analysis*** (CRA), ki temelji na omrežjih besedil. Za prikaz zmogljivosti CRA sta ustvarila in analizirala več omrežij. Med njimi tudi omrežje ***Reuters terror news***, ki ga sestavljajo vse Reutersove vesti v prvih 66 zaporednih dnevih, ki se nanašajo na dogodke povezane s terorističnim napadom na ZDA 11. septembra 2001.

To omrežje smo *Viszards* (skupina, ki se ukvarja s prikazi omrežij) lani izbrali za omrežje, ki ga bomo poskusili zanalizirati in prikazati na čim bolj različne načine. Rezultate smo predstavili na skupinski predstavitvi na Sunbelt XXII, International Sunbelt Social Network Conference, New Orleans, USA, 13-17. februar 2002.

Z Andrejem sva najprej združila 66 dnevnih CRA-jevih omrežij v eno samo Pajkovo časovno omrežje **`Days.net`** z $n = 13332$ točkami (različnimi besedami v vesteh) in $m = 243447$ povezavami, 50859 z vrednostjo večjo od 1. V omrežju ni zank.

Časovna omrežja v programu Pajek

```
*vertices  13332
 ...
 6842 "label"       [3,31,45,56,57,61]
 6843 "labor"       [2,6,20,25,31,43,53]
 6844 "laboratory"  [2,4,5,15,18,24,28,30,31,37,39-41,45-49,51-53,55,56,60]
 6845 "labour"      [21,22]
 6846 "lace"        [17,59]
 ...
10574 šcience"     [2,4,5,14-16,37,40-43,47,53,60]
10575 šcientific"  [15,49,53,60]
10576 šcientist"   [2,6,11,18,21,30,37,41,47,48,55-57,60,63]
10577 šcoff"       [13]
10578 šcope"       [4-7,10,15,17,28,53]
10579 šcorch"      [59]
10580 šcore"       [1-3,6,8,9,13,15,26,32,33,35,50,54]
10581 šcoreboard"  [29,50]
 ...
*edges
 ...
 8545   9227   4  [60]
 9227  10935   2  [60]
 1076  11885   2  [60]
 6844  10575   3  [60]
  417   6844   1  [60]
 9288  11741   2  [60]
 ...
```


Dvovrstna omrežja

Nekakšen *omrežni 'data mining'*.

Dvovrstno omrežje sestavljajo (U, V, A, w), kjer sta U in V ločeni množici točk, A je množica usmerjenih povezav z začetkom v U in koncem v V, ter je $w : A \to \mathbb{R}$ utež. Če utež ni določena, privzamemo, da je $w(u, v) = 1$ na vseh povezavah $(u, v) \in A$. Dvovrstno omrežje lahko predstavimo tudi s pravokotno matriko $\mathbf{A} = [a_{uv}]_{U \times V}$.

$$a_{uv} = \begin{cases} w_{uv} & (u, v) \in A \\ 0 & \text{otherwise} \end{cases}$$

Pri analizi dvovrstno omrežje pogosto prevedemo na prirejeno navadno (enovrstno) omrežje $(U, \mathbf{A}\mathbf{A}^T, w_1)$ ali/in $(V, \mathbf{A}^T\mathbf{A}, w_2)$, na katerem lahko uporabimo običajne pristope.

Dvovrstna omrežja nad velikimi množicami U in/ali V lahko zmanjšamo tako, da določimo razbitje množic U in/ali V in nato skrčimo skupine.

Dvovrstna omrežja ...

Pri množenju matrik se lahko opremo na različne polkolobarje: $(+, \cdot)$, $(\vee, \wedge)$ za $w = 1$, (max, min), (min, $+$) ... in tako dobimo različna prirejena navadna omrežja.

Tudi v primeru omrežij CRA je v ozadju dvovrstno omrežje (enote besedila, besede, vsebuje), ki je programsko predelano v navadno omrežje (besede, nastopata skupaj, pogostost).

Posplošene sredice

Omogočajo nam določiti pomembne (goste) dele omrežja.

Naj bo (V, E, w) omrežje, V množica točk, E množica povezav, in $w : E \to \mathbb{R}$ utež na povezavah. Za točko $v \in V$ in $C \subseteq V$ definiramo vrednost p

$$p(v; C) = \sum_{u \in N(v) \cap C} w(v, u)$$

kjer je $N(v)$ množica sosedov točke v. (Posplošena) *t-sredica* omrežja imenujemo največjo množico C, za katero za vse $v \in C$ še velja $p(v; C) \geq t$.

V omrežju ***Terror news*** je utež w pogostost sopojavljanja danih dveh besed (krajišč povezave). V tem primeru je t-sredica največje podomrežje, v katerem se je vsaka točka sopojavila v enotah besedila vsaj t krat z z neko drugo (ne nujno isto) enoto iz podomrežja.

Za določanje posplošenih sredic obstaja učinkovit postopek zahtevnosti $O(m \log n)$, ki je vgrajen v program **Pajek**.

Usklajevanje uteži

Kako razkriti zgodbe iz ozadja? Ena pot je odstranitev glavnih zgodb in analiza ostanka; druga pa usklajevanje uteži. Zaradi velike razlike v pogostosti pojavljanja posameznih besed, je nemogoče primerjati uteži na povezavah naravnost – no, z ustreznimi pretvorbami jih lahko poskusimo uskladiti, narediti primerljive.

$$\begin{aligned}
\text{Geo}_{ij} &= \frac{w_{ij}}{\sqrt{w_{ii} w_{jj}}} & \text{GeoDeg}_{ij} &= \frac{w_{ij}}{\sqrt{\deg_{ii} \deg_{jj}}} \\
\text{Input}_{ij} &= \frac{w_{ij}}{w_{jj}} & \text{Output}_{ij} &= \frac{w_{ij}}{w_{ii}} \\
\text{Min}_{ij} &= \frac{w_{ij}}{\min(w_{ii}, w_{jj})} & \text{Max}_{ij} &= \frac{w_{ij}}{\max(w_{ii}, w_{jj})} \\
\text{MinDir}_{ij} &= \begin{cases} \frac{w_{ij}}{w_{ii}} & w_{ii} \leq w_{jj} \\ 0 & \text{sicer} \end{cases} & \text{MaxDir}_{ij} &= \begin{cases} \frac{w_{ij}}{w_{jj}} & w_{ii} \leq w_{jj} \\ 0 & \text{sicer} \end{cases}
\end{aligned}$$

Usklajevanje uteži ...

Omrežja brez zank dopolnimo tako, da na diagonalo postavimo ustrezne uteži. Pri neusmerjenih omrežjih vrstično (ali stolpčno) vsoto

$$w_{vv} = \sum_u w_{vv}$$

pri usmerjenih pa neko srednjo vrednost vrstične in stolpčne vsote – na primer:

$$w_{vv} = \frac{1}{2}\left(\sum_u w_{vu} + \sum_u w_{uv}\right)$$

Običajno zahtevamo, da omrežje ne vsebuje osamljenih točk.

Usklajevanje uteži . . .

Če prerežemo usklajeno omrežje (V, E, w') na ravni t

$$E' = \{e \in E : w'(e) \geq t\}$$

dobimo (pod) omrežje $(V(E'), E', w')$, ki ima običajno večje število komponent – *otokov*, ki v našem primeru določajo posamezne *zgodbe*. Njihovo število je odvisno od praga t. Pogosto se pojavi veliko malih komponent. Za zanimivo zgodbo potrebujemo vsaj nekaj besed. Zato se omejimo na komponente, ki vsebujejo vsaj k točk. Primerni vrednosti pragov t in k določimo s pregledom porazdelitve vrednosti uteži in porazdelitve velikosti komponent.

Usklajevanje uteži je bilo prvič uspešno uporabljeno na dvovrstnem omrežju branosti slovenskih časopisov in revij.

Primer: Reuters terror news

```
[read network: Days.net]
Net/Transform/remove/multiple lines/sum values
Net/Components/Weak [1]
Info/Partition [OK][OK]
Operations/Extract from Network/Partition [1][1]
[save network: DaysAll.net ]
[select network: Days]
Operations/Extract from Network/Partition [1][1]
[save network: DaysCom.net ]
[select network: DaysAll]
Net/Partitions/Valued Core/First Threshold and Step/Input [0][25]
Operations/Extract from Network/Partition [20][99]
Draw
Options/Mark Vertices Using/Labels
Options/Lines/Different Widths
Layout/Energy/Kamada-Kawai/Free
[manually improve the layout]
Export/Options [3D Effects on Vertices: check, Gradients: radial]
Export/SVG/Line Values/Options/GreyScale
Export/SVG/Line Values/Options/Different Widths
Export/SVG/Line Values/Nested Classes  [Core500.htm][#10]
[select network: DaysAll]
Info/Network/Values of lines [#100]
Net/Transform/Remove/lines with value/less than [50]
Net/Components/Weak [2]
Operations/Extract from Network/Partition [1][99]
Draw
Layout/Energy/Fruchterman Reingold/2D
```


Primer: Reuters terror news . . .

```
[select network: DaysCom]
Operations/Extract from Network/Partition [1][99]
Net/Transform/Generate in Time/All [1][66][1]
[select network: in time 1]
Draw
[next - previous ...]
[select network: in time 1]
Export/SVG/Current and all Subsequent
Export/SVG/Line Values/Nested Classes [deg50][#15]
```

```
[select network: DaysAll]
Net/Partitions/Degree/Input
Partition/Make vector
Operations/Vector/Put loops
Net/Transform/2-mode to 1-mode/Normalize/Geo
Net/Transform/Remove/loops
[save network: GeoDeg.net ]
Info/Network/Line values [#100][No]
Net/Transform/Remove/lines with values/lower than [0.25][Yes]
Net/Components/Weak [6]
Operations/Extract from network/Partition  [1] [99]
Net/Components/Weak [1]
Draw/Partition
Layout/Energy/Kamada-Kawai/Free
Move/Grid [50][50]
```


Primer: Reuters terror news . . .

```
[select network   DaysAll.net ]
Net/Partitions/Valued Core/First Threshold and Step/Input [0][20]
Operations/Extract from network/Partition  [2] [999]
Vector/Create Identity Vector [3122]
Operations/Vector/Network*Vector [1]
Operations/Vector/Put loops
Net/Transform/2-mode to 1-mode/Normalize/MinDir
Net/Transform/Remove/Loops
[save network: MinDir.Net ]
Info/Network/Line values [#100][No]
Net/Transform/Remove/lines with values/lower than [0.25][Yes]
Net/Components/Weak [4]
Operations/Extract from network/Partition  [1] [999]
Net/Components/Weak [1]
Draw/Partition
Layout/Energy/Kamada-Kawai/Free
Move/Grid [50][50]
```

Naslovi

`http://vlado.fmf.uni-lj.si/pub/networks/doc/seminar/Solomon.pdf`

`http://vlado.fmf.uni-lj.si/pub/networks/doc/seminar/Solomon-4.ps`

`http://vlado.fmf.uni-lj.si/pub/networks/doc/seminar/SolomonURLs.htm`

`http://vlado.fmf.uni-lj.si/pub/networks/pajek/`

`vladimir.batagelj@uni-lj.si`