

Dinamični prikaz časovnih omrežij

Darko Brvar¹, Andrej Mrvar² in Vladimir Batagelj³

Univerza v Ljubljani

Univerzitetni podiplomski študijski program statistika¹

Fakulteta za družbene vede², Fakulteta za matematiko in fiziko³

darko.brvar@siol.net, andrej.mrvar@fdv.uni-lj.si, vladimir.batagelj@fmf.uni-lj.si

Povzetek

Časovna omrežja lahko prikažemo statično ali dinamično. Pri statičnem prikazu omrežje prikažemo s sliko ali zaporedjem slik, pri dinamičnem pa gre za zvezno prehajanje med časovnimi rezinami omrežja, kjer sledi prejšnjih slik omrežja postopno izginjajo. Statične prikaze časovnih omrežij lahko izdelamo s programskim paketom Pajek. V članku bomo predstavili program PajekToSvgAnim, ki temelji na slikovnem jeziku SVG in programskem jeziku Python in ki je programski dodatek programskemu paketu Pajek za dinamični prikaz časovnih omrežij. Prikazali bomo delovanje programa PajekToSvgAnim na nekaterih znanih primerih časovnih omrežij.

Abstract

DYNAMIC VISUALIZATION OF TEMPORAL NETWORKS

Temporal networks can be visualized statically or dynamically. While static visualization presents networks with pictures, dynamic visualization shows continuous passes of networks between time points with traces of previous network pictures. Static visualization can be produced using program package Pajek. We will present program PajekToSvgAnim, based on graphical language SVG and programming language Python, as addition to program package Pajek for dynamic visualization of temporal networks. Program PajekToSvgAnim will be demonstrated on some well-known examples of temporal networks.

1 Časovna omrežja

1.1 Definicija grafa in omrežja

Graf G je definiran kot par $G = (V, L)$, kjer je V množica točk in L množica povezav. Povezave so lahko usmerjene ali neusmerjene. Predpostavljamo, da je bralec seznanjen z osnovami teorije grafov (na primer [12]).

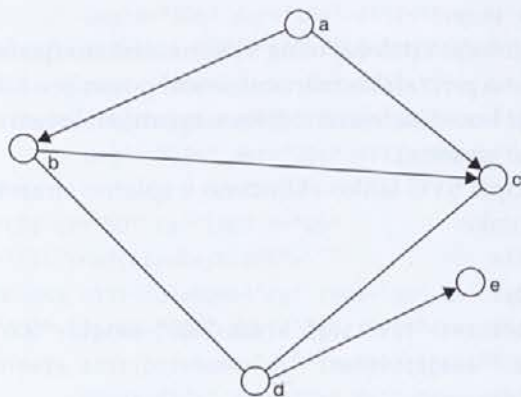
Graf postane omrežje, če vsebuje dodatne podatke o točkah in/ali povezavah. Omrežje N lahko torej definiramo kot četverico množic $N = (V, L, F_V, F_L)$, kjer je V množica točk, L množica povezav, F_V množica lastnosti točk in F_L množica lastnosti povezav.

Množica lastnosti točk F_V je množica funkcij $f: V \rightarrow X$, kjer množica X lahko predstavlja množico oznak

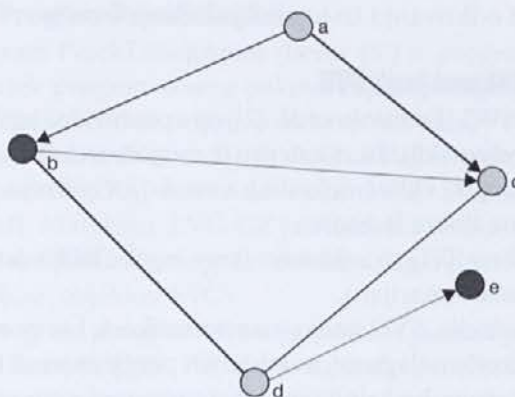
točk, razbitje točk ali množico številskih lastnosti točk. Na sliki lahko številsko lastnost prikažemo kot velikost točke ali njeno koordinato, imensko lastnost pa kot barvo, obliko lika ali kot oznako točke.

Množica lastnosti točk F_L pa je množica funkcij $g: L \rightarrow Y$, kjer je Y množica številskih lastnosti povezav. Na sliki jih prikažemo z izpisom vrednosti, debelino črte ali sivino.

Primer: Na sliki 1.1a je prikazan graf z množico točk $\{a, b, c, d, e\}$, množico usmerjenih povezav $\{(a, b), (a, c), (b, c), (d, e)\}$ in množico neusmerjenih povezav $\{(b, d), (c, d)\}$. Na sliki 1.1b pa je prikazan isti graf z imensko lastnostjo točk (razbitjem), predstavljeno z dvema barvama ter s številsko lastnostjo povezav,



Slika 1.1a: Primer grafa



Slika 1.1b: Graf z dodatnimi lastnostmi

predstavljeno z različnimi debelinami (barve niso razvidne zaradi črno-belega tiska, zato se članek nahaja tudi na spletni strani [2]).

1.2 Definicija časovnega omrežja

Časovno omrežje je omrežje, v katerem se prisotnost posameznih točk in povezav pa tudi njihove lastnosti spreminjajo skozi čas. Zato ga lahko definiramo na naslednji način:

Časovno omrežje N_T definiramo kot peterico množic $N_T = (V, L, F_V, F_L, T)$, kjer je V množica točk, L množica povezav, F_V množica lastnosti točk, F_L množica lastnosti povezav in T množica linearno urejenih časovnih trenutkov.

Točke in povezave, ki pripadajo časovnemu omrežju, niso nujno prisotne v vseh časovnih trenutkih. Pri tem velja logična omejitev, da je povezava prisotna v določenem časovnem trenutku, če sta v tem trenutku prisotni tudi obe njeni krajišči.

1.3 Časovna socialna omrežja

Pri analizi časovnih socialnih omrežij se zanimanje vrti okrog razumevanja, kako se omrežje razvija in spreminja skozi čas, in okrog iskanja načinov za razvoj modelov socialnih procesov, ki bi pomagali pojasniti opažene strukture (Doreian et al. [6]). Jedro zanimanja je torej dinamika medosebnih odnosov, ki je pomembna za razumevanje socialnega omrežja. Pri tem se pojavi problem, da se te dinamike ne da na zadovoljiv način prikazati s statičnim prikazom (Bearman, Everett [4]). Rešitev je dinamični prikaz časovnih omrežij, ki raziskovalcem odpira vrata na naslednjih področjih:

- pri odkrivanju značilnosti razvoja omrežja,
- pri spremljanju razvoja izbranih delov omrežja (tirov),
- pri odkrivanju izstopajočih delov omrežja.

2 Slikovni jezik SVG

Jezik SVG (Ferraiole et al. [7]) je označevalni jezik za opis vektorskih slik, s katerim je mogoče izdelati spletne strani, ki vključujejo slike z visoko ločljivostjo. Ima številne dobre lastnosti:

- slike v SVG so vektorske (brez izgube ločljivosti pri transformacijah),
- datoteke .SVG imajo manjšo velikost, kar pomeni hitrejše nalaganje, in lahko jih pregledujemo kar s spletnim brskalnikom, v katerega prej namestimo prosto dostopen dodatek za pregledovanje,

- v spletnem brskalniku deluje tudi iskanje nizov/besedil v sliki SVG,
- osnova jezika SVG je označevalni sestav XML, ki ima danes že vodilno vlogo pri izmenjavi najrazličnejših podatkov na spletu in tudi drugje.

2.1 Zgradba opisa SVG

Opis SVG lahko definiramo kot samostojno datoteko ali kot vgrajeno datoteko v spletni strani. Spodnji primer predstavlja samostojno datoteko .SVG:

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.0//EN"
    "http://www.w3.org/Graphics/SVG/1.1/DTD/
    svg11.dtd">
<svg xmlns="http://www.w3.org/2000/svg"
    width="300" height="300" x="0" y="0">
    ....
    ....
    ....
</svg>
```

Prva vrstica vsebuje najavo XML glede na to, da je jezik XML osnova jezika SVG. Druga in tretja vrstica vsebujeta povezavo z opisom 'slovnice' jezika SVG (angl. Document Type Definition ali krajše DTD), ki mora biti tako kot najava XML sestavni del vsake datoteke .SVG.

Četrta in peta vrstica vsebujeta značko SVG, ki označuje, da gre za opis SVG. Značka SVG ima naslednje lastnosti:

- **xmlns:**
obvezna lastnost, ki določa imenski prostor jezika SVG,
- **height in width:**
določata velikost okna SVG na zaslonu,
- **x in y:**
določata položaj okna SVG na zaslonu (jezik SVG ima privzeti koordinatni sistem postavljen tako, da je koordinatno izhodišče v zgornjem levem vogalu zaslona).

Opis SVG lahko vključimo v spletno stran tudi z datoteke:

```
<html>
<body>
<embed src="test.svg" width="500" height="500"
    type="image/svg+xml" />
</body>
</html>
```


To naredimo z uporabo značke EMBED, ki ji določimo naslednje lastnosti:

- **src:**
spletni naslov datoteke .SVG (če se nahaja v drugi mapi kot datoteka .HTML, je potrebno navesti celotno pot),
- **height in width:**
velikost vgrajenega okna SVG na spletni strani,
- **type:**
zvrst datoteke oblike MIME (programski standard, ki definira enostaven mehanizem za opisovanje vsebinskih tipov datotek, ki se pošiljajo prek spleta) – za datoteke SVG so predvidene vrednosti `image/svg`, `image/svg+xml` in `image/svg+xml`.

2.2 Dinamični prikaz objektov v jeziku SVG

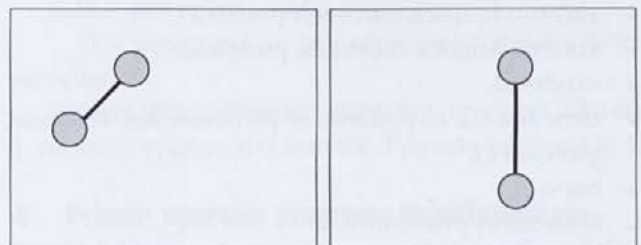
Jezik SVG ima za dinamični prikaz objektov na voljo več značk. Spoznali bomo osnovno značko `animate`, ki se uporablja za dinamični prikaz sprememb vrednosti lastnosti objektov SVG skozi čas. Oglejmo si primer grafa na dveh točkah, povezanih z neusmerjeno povezavo, ki se po določenem času počasi premakne na drug položaj (slika 2.1):

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.0//EN"
    "http://www.w3.org/TR/2001/PR-SVG-20010719/
    DTD/svg10.dtd">
<svg width="250" height="200"
    xmlns="http://www.w3.org/2000/svg" version="1.1">
  <rect x="2" y="2" width="246" height="196"
    fill="none" stroke="blue" stroke-width="2" />
  <line x1="50" y1="100" x2="100" y2="50"
    stroke-width="7" stroke="black">
    <animate attributeName="x1" from="50" to="150"
      begin="10s" dur="10s" fill="freeze"/>
    <animate attributeName="y1" from="100" to="150"
      begin="10s" dur="10s" fill="freeze"/>
    <animate attributeName="x2" from="100" to="150"
      begin="10s" dur="10s" fill="freeze"/>
  </line>
  <circle cx="50" cy="100" r="15"
    style="fill:red;stroke:black">
    <animate attributeName="cx" from="50" to="150"
      begin="10s" dur="10s" fill="freeze"/>
    <animate attributeName="cy" from="100" to="150"
      begin="10s" dur="10s" fill="freeze"/>
  </circle>
```

```
<circle cx="100" cy="50" r="15"
style="fill:red;stroke:black">
  <animate attributeName="cx" from="100" to="150"
    begin="10s" dur="10s" fill="freeze"/>
</circle>
</svg>
```

Iz primera lahko vidimo, da ima značka `animate` naslednje lastnosti:

- **attributeName:**
ime lastnosti nadrejenega objekta, ki se ji spreminja vrednost;
- **from in to:**
stara in nova vrednost te lastnosti;
- **begin:**
začetek dinamičnega prikaza spremembe vrednosti lastnosti;
- **dur:**
trajanje animacije;
- **fill:**
izbira, ki vpliva na to, ali nadrejeni objekt po koncu dinamičnega prikaza ohrani novo vrednost lastnosti (`freeze`) ali jo zamenja s staro vrednostjo (`remove`).



Slika 2.1: Dve točki in povezava med njima na začetku in na koncu dinamičnega prikaza

3 Program PajekToSvgAnim

Program `PajekToSvgAnim` (Brvar [5]) je programski dodatek programskemu paketu `Pajek` (Batagelj, Mrvar [3], [8], [10]) za pripravo dinamičnih opisov v SVG razvoja časovnih omrežij. Iz vhodne Pajkove projektne datoteke .PAJ izdelava datoteke .SVG, .SVG.GZ in .HTML (datoteka .SVG.GZ je stisnjena oblika GZIP datoteke .SVG, ki omogoča hitrejše nalaganje in hitrejši prikaz objektov SVG).

Program je napisan v programskem jeziku Python (Van Rossum [11]). Python je tolmačeni interaktivni objektni programski jezik in tako kot programski jezik Java omogoča izdelavo modulov, razredov, izjem

in dinamičnih podatkovnih tipov. Omogoča tudi izdelavo samostojno izvedljivih prevedenih programov.

Na začetku vhodne Pajkove datoteke se nahajajo podatki o celotnem omrežju, torej o vseh točkah in povezavah skupaj z njihovimi lastnostmi. Sledijo podatki o omrežjih v posameznih časovnih rezinah. Za ta omrežja so podani samo podatki o tistih točkah in povezavah, ki so v rezini dejansko prisotne.

Na koncu vhodne datoteke se nahaja še točkovno razbitje, pri katerem za vsako točko navedemo, ali je v prikazu njena oznaka vidna ali ne (vrednost 1: oznaka je vidna, vrednost 0: oznaka ni vidna).

Lastnosti točk in povezav v posameznih omrežjih so standardne lastnosti, ki so določene že v programskem orodju Pajek. Tako lahko točki določimo naslednje lastnosti:

- oznako,
- kooordinati x in y (in z),
- vrednost,
- obliko,
- razteg v smeri x,
- razteg v smeri y,
- barvo,
- interval(e) prisotnosti (npr. [1-3, 7, 12-15]).

Povezavi pa lahko določimo naslednje lastnosti:

- začetno krajišče (obvezen podatek),
- končno krajišče (obvezen podatek),
- vrednost,
- debelino (če ni podana, se privzame kar vrednost povezave),
- barvo,
- interval(e) prisotnosti.

Program PajekToSvgAnim upošteva naslednje posebnosti časovnih omrežij:

- zvezno spreminjanje koordinat točk,
- izginjanje točk in povezav in pojavljanje novih točk in povezav glede na to, ali je določena točka oziroma povezava prisotna v določenem časovnem trenutku,
- spreminjanje velikosti točk in debelin povezav brez izgube ločljivosti,
- prikaz večkratnih omrežij na istih točkah.

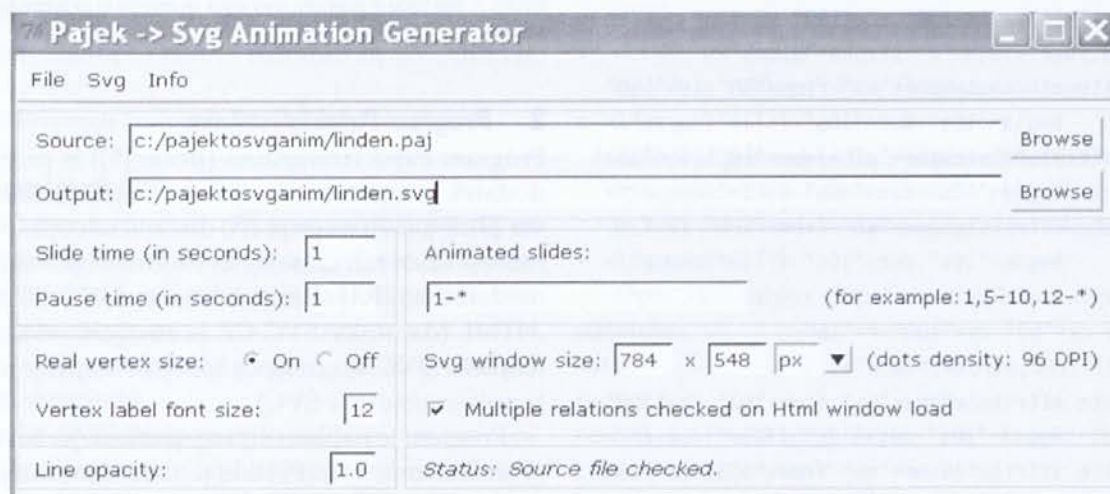
V program je vgrajenih več izbir, ki se lahko določajo v slikovnem uporabniškem vmesniku (slika 3.1). Poleg osnovnih izbir vhodne Pajkove datoteke in imena ter poti izhodne datoteke .SVG omogoča še naslednje izbire:

- dolžino trajanja dinamičnega prikaza prehoda omrežja iz ene rezine v drugo (Slide time),
- dolžino trajanja statičnega prikaza rezine pred dinamičnim prikazom prehoda v naslednjo rezino (Pause time),
- številski seznam dinamičnih prehodov med rezinami, ki naj se prikažejo (Animated slides).

Primer: Animated slides: 1, 5-10, 12-.*.

V tem primeru si prikazi omrežja sledijo v naslednjem vrstnem redu:

- dinamični prikaz prehoda od prve do druge rezine,
- statični prikazi od druge do pete rezine,
- dinamični prikazi prehodov med peto do enajsto rezino (poleg vmesnih statičnih prikazov),
- statična prikaza enajste in dvanajste rezine,
- dinamični prikazi prehodov med dvanajsto in zadnjo rezino (poleg vmesnih statičnih prikazov);



Slika 3.1: Izbira vhodne Pajkove datoteke v programu PajekToSvgAnim

- možnost izbire med dvema velikostima oblik točk: dejansko velikostjo, ki je podana v vhodni datoteki, in enotno velikostjo (Real Vertex Size On/Off),
- velikost pisave oznak točk (Vertex label font size),
- velikost okna SVG (Svg window size),
- možnost izbire označitve relacij v spletnem sestavku z vgrajeno sliko SVG (ob vsaki naložitvi sestavka) v primeru večkratnega omrežja (Multiple relations checked on Html window load),
- neprosojnost povezav (Line opacity).

Program najbolj preprosto uporabimo tako, da kar poženemo izdelavo datotek .SVG, .SVG.GZ in .HTML z ukazom `Svg → Generate`. Če je izdelava datotek uspešna, se prikaže ustrezno sporočilo. Dinamični prikaz si ogledamo tako, da odpremo izdelano datoteko .HTML, ki poleg okna SVG vsebuje še izbiro prikaza oznak točk in izbiro prikaza relacij v primeru večkratnega omrežja. Datoteka .HTML je povezana s stisnjeno datoteko .SVG.GZ zaradi hitrejšega prenosa izdelanih objektov v oknu SVG.

V nadaljevanju si oglejmo nekaj primerov, v katerih lahko uporabimo eno ali več od zgoraj naštetih izbir.

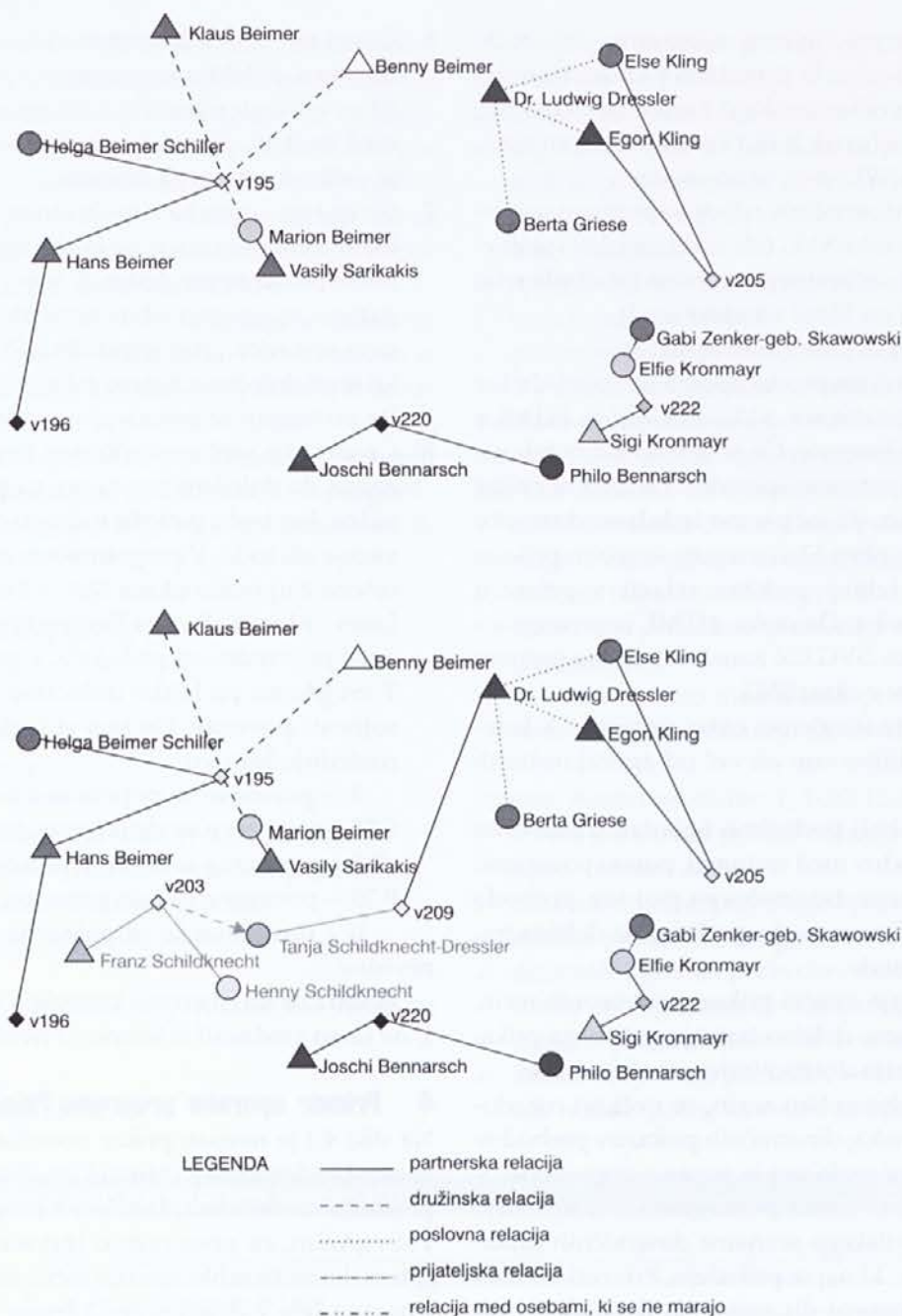
1. Če si želimo bolj podrobno ogledati dinamične prikaze prehodov med rezinami, potem povečamo dolžino trajanja dinamičnega prikaza prehoda omrežja iz ene rezine v drugo. Privzeta dolžina trajanja je 1 sekunda.
2. Če nas zanimajo statični prikazi posameznih rezin, potem povečamo dolžino trajanja statičnega prikaza rezin. Privzeta dolžina trajanja je 1 sekunda.
3. Če ima omrežje veliko rezin, se večkrat osredotočimo le na nekaj dinamičnih prikazov prehodov med izbranimi rezinami in ne na vse prehode. V tem primeru si lahko pomagamo z možnostjo določitve številskega seznama dinamičnih prikazov prehodov, ki naj se prikažejo. Privzeti seznam je 1-*, kar pomeni dinamični prikaz vseh prehodov.
4. Možnost izbire med dejansko velikostjo točk, ki je podana v vhodni datoteki, in enotno velikostjo za vse točke je možnost, ki jo poznamo iz programskega paketa Pajek. Uporabimo jo, ko nas dejanske velikosti točk ne zanimajo ali pa ko so točke prevelike za prikaz.
5. Možnost izbire velikosti pisave oznak točk je klasična možnost, ki jo tudi poznamo iz programskega paketa Pajek. Privzeta velikost pisave je 12.
6. Če želimo dinamični prikaz vključiti v svojo spletno stran, si lahko pomagamo z možnostjo določitve velikosti okna SVG. Pri tem lahko izbiramo med enotami cm, inch, px. Privzeta velikost okna je velikost celotnega zaslona.
7. Če imamo opravka z večkratnim omrežjem z velikim številom relacij, se lahko zgodi, da bo dinamični prikaz nepregleden. V tem primeru si pomagamo z možnostjo izbire označitve relacij v spletnem sestavku z vgrajeno sliko SVG, s katero lahko sami določimo, katero relacijo želimo prikazati. Po privzetem se prikažejo vse relacije.
8. Če omrežje vsebuje večkratne povezave, se lahko zgodi, da določene povezave na prikazu ne bodo vidne, ker bodo prekrite z drugo povezavo, ki povezuje isti točki. V programskem paketu Pajek je to rešeno z uporabo ukaza `Net → Transform → Sort-Lines → LineValues → Descending`, ki uredi povezave po vrednosti padajoče, v programu PajekToSvgAnim pa lahko določimo stopnjo neprosojnosti povezav. Pri tem si lahko pomagamo z naslednjo lestvico:
 - 1 – povezave so popolnoma neprosojne,
 - 0.75 – povezave so delno prosojne,
 - 0.5 – povezave so srednje prosojne,
 - 0.25 – povezave so zelo prosojne,
 - 0 – povezave so popolnoma prosojne, torej nevidne.
 Seveda lahko izberemo katerokoli vrednost od 0 do 1, ne samo vrednosti iz lestvice. Privzeta vrednost je 1.

4 Primer uporabe programa PajekToSvgAnim

Na sliki 4.1 je narisani prikaz omrežja zgodb nadaljevanke Lindenstrasse (Mutzel [9], Batagelj [11]) z nepremičnimi točkami, izdelan s programom PajekToSvgAnim, za prvo rezino (prva zgodba) in malo zatem, ko se že rahlo opazijo točke in povezave, ki so prisotne šele v drugi rezini (druga zgodba). Delno prosojne na novo prihajajoče točke in povezave (npr. točka z oznako v209 in povezava od te točke do točke z oznako Dr. Ludwig Dressler) postajajo tem bolj vidne, čim bolj se bliža druga rezina. Več takih spletnih prikazov (v barvah) je dosegljivih na spletni strani (Batagelj [2]).

5 Sklep

Zaradi povečanega zanimanja za raziskovanje časovnih omrežij in zaradi dejstva, da prikaz omogoča boljši



Slika 4.1: Prikaz prve rezine omrežja Lindenstrasse z nepremičnimi točkami, izdelan s programom PajekToSvgAnim, in med prehodom

vpogled v značilnosti omrežij tako pri začetnem raziskovanju omrežij kot tudi pri predstavitvi rezultatov raziskav, se je v zadnjem času pojavila potreba po dinamičnem prikazu omrežij.

V ta namen je bil izdelan PajekToSvgAnim kot programski dodatek programskemu paketu Pajek za dinamični prikaz časovnih omrežij.

S pomočjo programa PajekToSvgAnim lahko izdelamo zvezne prehode časovnih omrežij med časovnimi rezinami. Pri posameznem prehodu omrežja iz ene rezine v drugo se vidijo sledi prejšnjih slik omrežja, torej sledi tistih točk in povezav omrežja, ki v naslednji rezini niso več prisotne. To nam omogoča, da lahko bolj natančno sledimo spremembam omrežja skozi

čas kot doslej, ko smo imeli na voljo le statične slike omrežja.

Program PajekToSvgAnim ima za različne potrebe uporabnikov vgrajenih kar nekaj izbir, ki se lahko določajo v slikovnem uporabniškem vmesniku. Če si želimo npr. bolj podrobno ogledati dinamične prikaze prehodov med rezinami, povečamo dolžino trajanja dinamičnega prikaza prehoda omrežja iz enega rezine v drugo. Če želimo dinamični prikaz vključiti v svojo spletno stran, si lahko pomagamo z možnostjo določitve velikosti okna SVG. Če pa imamo opravka z večkratnim omrežjem z velikim številom relacij, uporabimo možnost izbire označitve relacij v spletnem sestavku z vgrajeno sliko SVG, s katero lahko sami določimo, katere relacije želimo prikazati.

6 Viri in literatura

- [1] BATAGELJ, Vladimir: *Pajek Datasets*.
[URL: <http://vlado.fmf.uni-lj.si/pub/networks/data/>].
- [2] BATAGELJ, Vladimir: *Pajek to SVG Anim*.
[URL: <http://vlado.fmf.uni-lj.si/pub/networks/pajek/svgAnim/>].
- [3] BATAGELJ, Vladimir, MRVAR, Andrej: *PAJEK 1.02, Program for Large Network Analysis*.
[URL: <http://vlado.fmf.uni-lj.si/pub/networks/pajek/>].
- [4] BEARMAN, P., EVERETT, K.: *The Structure of Social Protest*. *Social Networks* 15, 1993, str. 171.
- [5] BRVAR, Darko: *Dinamični prikaz časovnih omrežij*. Magistrsko delo. Ljubljana, 2005.
- [6] DOREIAN, P., KAPUSCINSKI, R., KRACKHARDT, D., SZCZYPULA, J.: *A Brief History of Balance Through Time*. *Journal of Mathematical Sociology* 21(1-2), 1996, str. 113.
- [7] FERRAILOLO, J., JUN, F., JACKSON, D.: *Scalable Vector Graphics (SVG) 1.1 Specification*. W3C Recommendation.
[URL: <http://www.w3.org/TR/2003/REC-SVG11-20030114/>]
- [8] MRVAR, Andrej: *Analiza in prikaz velikih omrežij*. Doktorska disertacija. Ljubljana, 1999.
- [9] MUTZEL, P.: *GD99 contest: Lindenstrasse network*, 1999.
[URL: <http://kam.mff.cuni.cz/conferences/GD99/contest/graphs/A.html>]
- [10] de NOOY, Wouter, MRVAR, Andrej, BATAGELJ, Vladimir: *Exploratory Social Network Analysis with Pajek*. New York: Cambridge University Press, 2005
- [11] VAN ROSSUM, G.: *Python*.
[URL: <http://www.python.org/2.4/>].
- [12] WILSON, Robin. J., WATKINS, John J.: *Uvod v teorijo grafov*. Sigma, št. 63, DMFA Slovenije, Ljubljana, 1997.

Mag. Darko Brvar je univerzitetni diplomirani inženir matematike. Konec leta 2005 je zaključil podiplomski študij statistike, smer družboslovna statistika. Ukvarja se z dinamičnim prikazom časovnih omrežij. Temo magistrskega dela je predstavil širši strokovni javnosti na Dnevih slovenske informatike aprila 2006.

Doc. dr. Andrej Mrvar je zaposlen na Fakulteti za družbene vede, kjer predava predmete s področja informatike in analize podatkov. Ukvarja se predvsem s teorijo grafov, algoritmi na grafih in omrežjih in analizo podatkov. Je soavtor programa za analizo in prikaz velikih omrežij Pajek. Letos je pri založbi Cambridge University Press izšla monografija W. de Nooy, A. Mrvar, V. Batagelj: *Exploratory Social Network Analysis with Pajek*.

Dr. Vladimir Batagelj je zaposlen kot redni profesor na Fakulteti za matematiko in fiziko, kjer predava predmete s področja diskretne in računalniške matematike. Ukvarja se predvsem s teorijo grafov, algoritmi na grafih in omrežjih, kombinatorično optimizacijo, analizo podatkov in uporabo informacijske tehnologije v izobraževanju. Je član več domačih in mednarodnih strokovnih združenj. Sam ali kot soavtor je napisal več visokošolskih in srednješolskih učbenikov, priročnikov in poljudnih del, številne znanstvene, strokovne in poljudne članke. Je soavtor programa za analizo in prikaz velikih omrežij Pajek.