INFOTEH-JAHORINA Vol. 14, March 2015.

Particionisanje grafa - problem MIN-CUT

Rava Filipović
ORAO a.d.
Bijeljina, Republika Srpska, BiH
filipovicrava@yahoo.com

Sadržaj—**U ovom radu je opisan razvoj efikasnih i efektivnih postupaka particionisanja članova neke društvene mreže, sa posebnim osvrtom na rješavanje MIN-CUT problema optimizacije na grafovima, koji koristi Ncut kao funkciju cilja. Za rješavanje problema, predložena je Metoda promjenljivih okolina. Ovaj rad je implementiran u okviru rješavanja problema podjele u klastere tako da normalizovan presjek, kao jedan od kriterijuma klasterovanja grafa, optimizuje funkciju cilja u društvenim mrežama.**

Ključne riječi-klasteri; optimizacija; heuristike;

I. UVOD

Kako se mnogi problemi iz realnog života mogu svesti na grupisanje čvorova u grafu, tako problemi particionisanja grafa imaju široku primjenu pri analizi socijalnih mreža. Prirodan način podjele objekata u grupe ima za cilj smanjivanje broja veza između različitih grupa. Težište ovog istraživanja je razvoj efikasnih i efektivnih postupaka klasterovanja članova neke društvene mreže. Za rješavanje problema predložena je metaheuristika [1] promjenljivih okolina. Ova metoda računa presjek grafa sa najmanjim brojem prelaznih ivica čiji krajevi su u različitim klasterima.

II. PARTICIONISANJE U DRUŠTVENIM MREŽAMA

Particionisanje [2], kao jedna od aktivnosti istraživanja podataka, rješava optimizacione probleme grupisanja objekata na osnovu nekih veza (relacija) koje postoje između njih. Pruža odličnu podršku procesu otkrivanja znanja. Postoji veliki broj definicija particija u literaturi, na primjer [1], [2]. Sve one nose isti pojam podskup grafa (ovde shvata kao graf odnosa prijateljstava korisnika društvene mreže [3]) u kome je koncentracija linkova između čvorova (susjeda) unutar klastera znatno veća nego koncentracija linkova između klastera. Particionisanje predstavlja jedan tip postupka klasterovanja, pri čemu je particija neprazan skup i presjek particija je prazan skup.

Pojam particija je najlakše razumjeti na primjerima kroz koje smo svi prošli i kroz koje svakodnevno prolazimo. Na primjer, svako ima prijatelje iz osnovne, srednje škole, fakulteta ili iz svakodnevnog života. Uopšte nije slučajno što smo odabrali baš njih za svoje prijatelje. Očigledno da postoje zakonitosti koje definišu koga možemo imati nekog za prijatelja. Te zakonitosti se svakako odnose na sve parametre koji utiču na sklapanje nekog prijateljstva, uključujući obe potencijalne strane, ali i okruženje, vrijeme i sve ostale uticaje u kojima se prijateljstva dogadjaju. Ako na primjer spoznamo težinsku uticajnost pojedinih parametara, onda možemo odgovarujućom analizom, sa znatno manjim brojem parametara, stvoriti grupe (particije) u okviru kojih je vjerovatnoća sklapanja prijateljstava značajno povećana u odnosu na slučajno odabranu grupu. Dakle te grupe predstavljaju stvarne grupe ljudi u svakodnevnom životu. Potrebno ih je samo razotkriti. Ako ih razotkrijemo, neće li to donijeti mnoge blagodeti, olakšice i pogodnosti svakom pojedincu ponaosob?

U ovom istraživanju je razmotren graf[4] G=(V,E), sa V skup čvorova i E skup ivica, koji se koristi za predstavljanje društvenih mreža. Čvorovi se grupišu u klastere uz kriterij da se klasteri ne preklapaju i da je velika razlika između unutrašnje i spoljašne gustine. Cilj je pronaći grane presjeka, koje odvajaju graf na K disjunktnih (odvojenuh) podskupova. U ovom slučaju, klaster predstavlja skup osoba (individua, pojedinaca) sa gustom relacijom prijateljskih veza unutar i rijetkom broju prijateljstava između klastera. Analiza društvene mreže posmatra društvene relacije u terminima čvorova i veza. Čvorovi su individualni akteri unutar mreže, a veze su njihovi međusobni odnosi. Otkriće tijesno povezanih klastera u ovim mrežama je od fundamentalnog i praktičnog interesa. U svojoj najjednostavnijoj formi društvena mreža je mapa svih relevantnih veza izmedu čvorova koji se proučavaju.

III. NORMALIZOVAN PRESJEK

Normalizovan presjek ([4], [5], [9], [12]) je jedan od najpoznatijih kriterijuma postupka klasterovanja grafa. Presjek klastera je definisan kao ukupna težina ivica pri čemu je jedan kraj u jednom klasteru, a drugi kraj u nekom od preostalih klastera (particija). Normalizovan presjek klastera se definiše

1Metaheuristike su opšti okvir za izgradnju heuristika za rješavanje problema kombinatorne i globalne optimizacije.

2Particija grafa misli se na particiju skupa njegovih čvorova.

3Društvena mreža (takođe i socijalna mreža) je društvena struktura sastavljena od pojedinaca (ili organizacija) koji se nazivaju "čvorovi", a koji su povezani jednim ili više specifičnih tipova „veza“, kao što su: prijateljstvo, vizije, ideje, finansijski interes, srodstvo, zajednički interes, finansijska razmena, nedopadanje, ili odnosi poverenja , znanja ili prestiža.

4downloaded from http://userweb.cs.utexas.edu/users/dml/Software/graclus.html kao suma odnosa ukupnog broja ivica (čiji je jedan kraj u jednom klasteru a drugi kraj u nekome od preostalih klastera) i broja ivica klastera (pri čemu je jedan kraj u klasteru, a drugi kraj u istom ili u nekom od preostalih klastera). Odnosno, normalizovan presjek je suma odnosa broja susjeda čvorova klastera u svim ostalim klasterima i broja susjeda u svim klasterima. Za njegovo rješavanje predložena je Metoda promjenljivih okolina. Ona nastoji da prevaziđe lokalne optimume mijenjanjem strukture okolina. Njen cilj je da se dobije približno dobro rješenje za relativno kratko vrijeme, uz kriterij da se normalizovan presjek optimizuje. Kriterij normalizovanog presjeka [12] minimizira funkciju cilja, pri čemu se formalno definiše:

$$\text{Ncut}(Pk=C1,C2,\ldots,Ck) = \sum_{l=1}^{k} \frac{\sum_{i\,:\,vi\,\in\,cl} \sum_{j\,:\,vj\,\notin\,Cl} wij}{\sum_{i\,:\,vi\,\in\,cl} si} \qquad (1)$$

gdje je $S_i = \sum_{j=1}^{n} w_{ij}$.

IV. METODA PROMJENLJIVIH OKOLINA (VNS)

Metoda promjenljivih okolina je metaheuristika bazirana na principu lokalnog pretraživanja. Njen osnovni okvir se koristi za izgradnju heuristika, za potrebe rješavanja problema kombinatorne i globalne optimizacije. Prije uvođenja VNS metaheuristike treba navesti neke osnovne pojmove optimizacije. Optimizacioni problem može biti formulisan kao:

$$\min\{f(x)\} \mid x \in X \subseteq S \qquad (2)$$

gdje je S prostor rješenja, X moguć skup rješenja, x moguće rješenje i f realna vrijednost objektivne funkcije. Rješenje x*□X je globalni optimum, ako je f(x*)≤f(x), ∀x□X.

Ideja ove metode je sistematska promjena okolina u okviru lokalnog pretraživanja. Ova metoda polazi od datog početnog (inicijalnog) rješenja x u čijoj okolini se na slučajan način (postupkom razmrdavanja) bira susjedno (slučajno) rješenje x'. Ovo slučajno rješenje postaje polazno rješenje za lokalno pretraživanje. Slučajno odabrano rješenje iz okoline trenutnog rješenja se poboljšava tehnikom lokalnog pretraživanja. Prostor dopustivih rješenja se pretražuje na slučajan način uz primjenu principa lokalnog pretraživanja. Prelazi se u bolje rešenje x'' i to u prvo na koje naiđe, tj. metoda je *First Improvement* (FI) karaktera. Ako se tako ne dobije bolje rješenje od trenutnog struktura trenutnog rješenja se mijenja i postupak ponavlja. Lokalno pretraživanje nastoji pronaći bolje rješenje. Glavna ideja je sledeća: ukoliko u okolini trenutnog rješenja nema boljeg rješenja pretraživanje se nastavlja u novoj okolini trenutno najboljeg rješenja. Ukoliko i tada se ne pronađe bolje rješenje, prelazi se u sljedeću okolinu, itd. sve dok se ne dođe do maksimalno unaprijed definisane okoline. Međutim, ukoliko u nekoj od okolina se pronađe bolje rješenje pretraživanje ponovo počinje od prve okoline. Složenost pretraživanja okoline se smanjuje generisanjem slučajnog rješenja u čijoj okolini se iterativno traži bolje rješenje.

Slično drugim metaheuristikama, metoda promjenljivih okolina zahtjeva tri parametra: t_{min}, t_{step} i t_{max}. Oni kontrolišu proces promjene okolina. Jednostavni su za razumijevanje i od velikog su značaja. Osnovni parametar VNS metode je t_{max} (maksimalan broj okolina). U ovom slučaju t_{min} definiše inicijalnu okolinu i $t_{min}=t_{step}$ (vrijednost povećanja indeksa okoline). Parametri t_{min} i t_{step} su sa značenjem da se ne polazi iz okoline t=1, već unaprijed definisane vrijednosti t_{min} tako da se indeks okoline ne uvećava za 1 nego za t_{step}.

Opisani koraci, primijenjene VNS metode, mogu se ilustrovati pseudokodom na sledeći način:

Inicijalizacija. Izabrati početno rješenje $x \in X$;
definisati kriterijum zaustavljanja; STOP = 0.
Ponavljaj
1. $t = t_{min}$;
2. **Ponavljaj**
   (a) *Razmrdavanje.* Generisati slučajno rješenje x' u t-toj okolini od x, ($x' \in N_t(x)$);
   (b) Lokalno pretraživanje *LS.* Primjeniti neku proceduru lokalnog pretraživanja počev od x';
   označiti sa x'' dobijeni lokalni minimum;
   (c) *Provjera rješenja.* Ako je lokalni minimum bolji od trenutnog minimuma, preći u to rješenje ($x = x''$);
   nastaviti od novog početnog rješenja u okolini N_1 ($t = t_{min}$);
   inače preći u sledeću okolinu, tj. $t = t + t_{step}$.
   (d) *Provjera završetka.* Ako je zadovoljen kriterijum zaustavljanja, STOP = 1.
   dok nije $t == t_{max}$ ili STOP == 1;
dok nije STOP == 1;

V. RJEŠENJE PROBLEMA

A. Definisan problem klasterovanja

Proučavan problem klasterovanja se može definisati na sledeći način: Dat je skup od N osoba i matrica prijateljstava između osoba. Rasporediti datih N osoba u K klastera tako da se minimizira funkcija Ncut (1).

B. Podjela u klastere

Skup od N osoba (datoteka čvorova) se particioniše u K disjunktnih podskupova, koji predstavljaju upravo tih K particija. Sve dok datoteka čvorova se ne isprazni radi se sledeće. Na početku algoritma puni se redom K klastera sa K slučajno odabranih čvorova, koji se uklanjaju iz datoteke čvorova. Potom se primjeni strategija smještanja slučajnog čvora u slučajni klaster. Pri čemu treba da bude zadovoljen uslov kod koga je količnik (čvor stepen prema jednom klasteru/trenutni broj čvorova u tom klasteru) najveći. Ukoliko je količnik nula onda se slučajno odabran čvor smješta u slučajno odabran klaster. Dobijeno rješenje predstavlja inicijalno (početno) rješenje, neophodno za primjenu metode promjenljivih okolina.

C. Postupak razmrdavanja

Postupak razmrdavanja mijenja strukturu inicijalnog (početnog) rješenja i u njegovoj okolini generiše slučajno rješenje. Slučajno rješenje se dobije na slučajan način u t-toj okolini početnog rješenja da bi se izbjeglo ponovno vraćanje u istu tačku. Na početku se na slučajan način odabere par klastera. U prvom klasteru se odabere na slučajan način jedan čvor. Ako slučajno odabran čvor nema susjeda u drugom slučajno odabranom klasteru algoritam se vraća na početak. Takođe, ako je broj susjeda slučajno odabranog čvora u drugom klasteru manji od broja susjeda u prvom klasteru postupak se ponavlja. Vjerovatnoća premještanja čvora iz jednog u drugi klaster treba da zadovolji uslov (3):

$$d_3<d_1/(d_1+d_2) \quad (3)$$

pri čemu je d_3 slučajno odabran broj iz opsega (0,1), d_1 broj međuklasterskih veza i d_2 je broj veza slučajno odabranog čvora u slučajno odabranom klateru. Ukoliko uslov (3) nije ispunjen ignoriše se prebacivanje čvora. Pri tome se realizuju putanje, koje se oslobode i pokušaju se realizovati putanje koje prethodno nisu bile realizovane. Na taj način se formiraju rješenja koja čine okolinu početnog rješenja. Postupak razmrdavanja se radi t puta. Veličina parametra t odgovara veličini indeksa okoline. Na kraju, računa se vrijednost funkcije cilja Ncut i pamti dobijeno slučajno rješenje. Ovo rješenje je neophodno za postupak lokalnog poboljšanja, čime će se nastojat poboljšat trenutno najbolje (početno) rješenje. Za slučaj prve okoline znači da će se osloboditi jedna realizovana putanja. Zatim slijedi lokalno pretraživanje i to sa strategijom prve popravke (First Improvement), koja se zaustavlja kada se dobije prvo bolje rješenje. Ova strategija se koristi iz razloga da se ne troši suviše mnogo vremena kako se ne bi istraživala čitava okolina slučajnog rješenja. Samo u krajnjem slučaju se pretražuje čitava okolina slučajnog rješenja, tj. ako se ne naiđe ni na jedno poboljšanje ili se pretraživanje zaustavlja ako istekne unaprijed određeno vrijeme.

D. Lokalno pretraživanje

Za redom odabran čvor, iščita se trenutni klaster slučajnog rješenja. Ako su trenutni i redom odabran klaster različiti računa se njihova funkcija cilja Ncut. Suma Ncut pojedinačnih vrijednosti (trenutnog i redom odabranog klastera) se oduzmu od ukupne vrijednosti funkcije cilja slučajnog rješenja. Čvor se premjesti iz trenutnog klastera u redom odabran klaster. Računa se vrijednost promjene funkcije cilja Ncut pojedinačnih (trenutnog i redom odabranog) klastera. Ova vrijednost se doda na trenutno izmijenjenu vrijednost Ncut-a slučajnog rješenja. Nova vrijednost slučajnog rješenja se pamti i predstavlja novo rješenje x" (lokalni optimum). Ako je novo rješenje bolje od trenutno najboljeg f(x")<f(x) bolje rješenje mijenja trenutno najbolje. Indeks okoline se postavlja na vrijednost t_{min}. Lokalno pretraživanje se nastavlja u okolinu novog najboljeg rešenja. Trenutni klaster postaje onaj u koji je čvor premješten. Vrijednost Ncut trenutnog klastera postaje Ncut klastera, u koji je prešao čvor. Postupak se nastavlja izborom sledećeg klastera. U slučaju lošijeg rješenja f(x")≥f(x) indeks okoline se poveća. Postupak lokalnog pretraživanja se vraća na postupak razmrdavanja. Ovim se proširuje okolina trenutno najboljeg rješenja i u njoj se ponovo bira slučajno rješenje.

Treba napomenuti da prilikom premještanja čvorova iz trenutnog klastera, u postupku razmrdavanja i postupku lokalnog pretraživanja, se vodi računa da se ne isprazni klaster (iz koga se premješta).

E. Kriterijum zaustavljanja

Kriterijumi zaustavljanja rada VNS algoritma je dostignuta unaprijed definisana maksimalna vrijednost indeksa okoline ili unaprijed definisano maksimalno dozvoljeno vrijeme izvršavanja.

F. Testni rezultati

Algoritam je implementiran u C#. Za potrebe testiranja korišteni su (neki) grafovi[5]. Broj čvorova i grana za svaki graf (skup podataka) je dat u Tabeli I.

Ograničenja postavljena u toku rada algoritma su ukupno vrijeme izvršavanja jednog VNS-a 1000 sec i definisani parametri okolina (4):

$$t_{min} = \min\{20, (n/100) + 1\},$$
$$t_{step} = \min\{20, (n/100) + 1\},$$
$$t_{max} = \min\{200, (n/5) + 1\}. \quad (4)$$

TABELA I. KARAKTERISTIKE GRAFOVA

Skup podataka	Broj čvorova i grana grafa	
	|*V*|	|*E*|
add32	4960	9462
finance256	37376	130560
gupta2	62064	2093111
memplus	17758	54196
pcrystk02	13965	477309
rajat10	37376	130560
ramage02	62064	2093111

Testni rezultati su ilustrovani primjerom i prezentovani u Tabeli II, koji daju pregled najboljih rješenja za pet prolaza VNS-a za datu instancu (skup podataka) i definisan broj klastera. Za ilustrovan primjer dobijenih popravljenih rješenja korištena je testna instanca[6] V=4960 čvorova i definisan broj klastera je K=128. Svako pojedinačno rješenje predstavljeno distribucijom cijelih brojeva (čvorova) po klasterima je zbog dozvoljenog prostora neizvodljivo prikazati, ali je dat prikaz dobijenih vrijednosti funkcije cilja Ncut i ukupno utrošenog vremena tokom popravljanja rješenja:

```
Ncut x  = 52.706265034515627460230334244
Vreme je 0, 0, 4, 461
tmin = 20
XBEST = 52.656619259479408558071744739
Vreme je 0, 0, 6, 442
XBEST = 52.127022336434409531280324856
Vreme je 0, 0, 10, 732
...
XBEST = 48.497775447734873497575007533
Vreme je 0, 0, 40, 453
XBEST = 48.480783162130917873498659111
Vreme je 0, 0, 40, 515
.....
XBEST = 36.722957862170804823590847544
Vreme je 0, 2, 8, 959
```

5downloaded from http://userweb.cs.utexas.edu/users/dml/Software/graclus.html

6 Instanca - konkretan test primjer.

```
XBEST = 36.398397381002631677398782760
Vreme je 0, 2, 13, 483
...
XBEST = 21.022702977315322654779164249
Vreme je 0, 6, 51, 806
XBEST = 20.875479903281939385408065867
Vreme je 0, 7, 1, 978
XBEST = 20.757583954150356077295778942
Vreme je 0, 7, 13, 615
...
XBEST = 19.533707619345627085742687014
Vreme je 0, 19, 3, 874
XBEST = 19.526480145247569963682325111
Vreme je 0, 19, 26, 452
XBEST = 19.487557519439766697968639464
Vreme je 0, 20, 18, 499
XBEST = 19.475289442353760379861464475
Vreme je 0, 20, 22, 210
XBEST = 19.440327998491482270120907777
Vreme je 0, 20, 40, 508
XBEST = 19.123201246910844500593759172
Vreme je 0, 20, 51, 104
```

TABELA II. TESTNI REZULTATI METODE VNS

Instanca	**Rezultati za def. broj klastera**	
	K	***SVNS***
add32	2	0.120
	4	0.390
	8	0.933
	16	1.988
	32	2.515
	64	4.164
	128	19.123
	256	48.276
finance256	2	0.124
	4	0.690
	8	1.040
	16	2.070
	32	3.349
	64	7.390
	128	21.230
	256	43.664
gupta2	2	0.320
	4	2.488
	8	7.540
	16	14.046
	32	23.343
	64	48.398
	128	99.120
	256	212.560

G. Pregled stanja riješenosti ovog problema u svijetu

Metaheuristike kao opšti okviri sa stohastičkim pretragama su često korišćeni u optimizaciji (poboljšanju) rješenja dobijenih od strane drugih heuristika. Normalizovan presjek kao jedan od kriterijumima klasterovanja grafa imao je primjenu u rješavanju problema optimizacije na grafovima, u društvenim mrežama. Neki od predloženih pristupa su spektralna metoda grupisanja ([4], [7]) i višestepeni pristup od strane Dhillon, Guan i Kulis [10]. Njihov cilj je bio da se dobiju dobra rješenja za relativno kratko vrijeme za velike instance. U radu [12] predložena je primjena VNS heuristike, koja koristi dobijeno inicijalno rješenje prethodno navedenog istraživanja. U rješavanju sličnog problema implementirani su testni rezultati Tabela III (prethodno navedenih istraživanja), koji će biti referentni za ovo istraživanje.

TABELA III. TESTNI REZULTATI

Instanca	**Rezultati za def. broj klastera**		
	K	***Graclus***	***VNS***
add32	2	0.002	0.002
	4	0.012	0.012
	8	0.059	0.059
	16	0.215	0.214
	32	0.800	0.766
	64	2.629	2.441
	128	13.141	10.668
	256	40.463	34.808
finance256	2	0.002	0.002
	4	0.010	0.010
	8	0.040	0.040
	16	0.353	0.347
	32	1.367	1.349
	64	5.527	5.349
	128	16.517	15.960
	256	40.194	38.664
gupta2	2	0.010	0.010
	4	1.183	1.183
	8	4.549	4.546
	16	9.049	9.046
	32	23.017	19.344
	64	48.944	42.398
	128	106.042	90.100
	256	213.340	194.560

H. Naučni doprinos

Najvažniji novi rezultati dobijeni istraživanjem prikazanim u ovom radu su:

- Definisana varijanta problema podjele u klastere koja je realnija (više odgovara praktičnim potrebama), ali zato kompleksnija za rješavanje.
- Definisanje odgovarajuće funkcije cilja.
- Definisan nov način kodiranja prezentacije rješenja cijelim brojevima.
- Grupisanje ili izdvajanje sličnih objekata iz neke veće cjeline u manje grupe, čime se pojednostavljuje analiza podataka.
- Definisanje broja i strukture okolina za tekuću varijantu problema.
- Razvoj efikasnih i efektivnih postupaka klasterovanja.

Glavni doprinos je razvoj metode promjenljivih okolina za prethodno uvedenu varijantu problema klasterovanja. Razvoj ovog algoritma je zahtjevao da se razvije:

- Novi način predstavljanja potencijalnih rješenja.
- Postupak za formiranje inicijalnih rješenja.
- Postupak razmrdavanja okoline trenutnog rješenja.
- Redosljed po kome se vrši pretraživanje.
- Strategija promjene okoline.
- Postupak proširenja okoline trenutnog rješenja.
- Razvijen postupak lokalnog pretraživanja.
- Kriterijum zaustavljanja.

Realizovani algoritam samostalno razvijen u zadovoljavajućem vremenu pronalazi izuzetno dobru podjelu u klastere. Kao što se može vidjeti iz eksperimentalnih rezultata, predložen metod promjenljivih okolina je veoma uspješan pri rješavanju problema podjele u klastere i namijenjen je rješavanju problema velikih dimenzija [12]. Zbog svega gore navedenog, naučno istraživanje opisano u ovom radu daje doprinos oblastima kombinatorne optimizacije i lokacijskih problema.

ZAKLJUČAK

S obzirom na skup entiteta, klaster analiza ima za cilj pronalaženje podskupova (klastera). Kao i mnogi tipovi grupisanja i kriterijumi za homogenost ili odvajanje su od interesa. Klaster analiza bavi se problemom pronalaženja podskupova nazvanih klasteri, u okviru takvog skupa. Obično, klasteri moraju da budu homogeni i/ili dobro razdvojeni. Homogenost znači da entiteti u istom klasteru liče jedan drugom, dok u suprotnom slučaju subjekti u različitim klastera se razlikuju jedan od drugog.

U radu je prikazan novi algoritam, koji koristi kriterijum min-cut (normalizovan presjek) da optimizuje performanse podjele grafa u klastere. Primijenjena metoda promjenljivih okolina ne prati putanju, već istražuje sve udaljenije okoline u odnosu na trenutno rješenje i odabire novo rješenje ako i samo ako je pronađeno rješenje bolje od trenutnog. Na ovaj način će mnoge poželjne karakteristike trenutnog rješenja biti prenesene u novo rješenje. Nad novim rješenjem se dodatno provodi lokalno pretraživanje kako bi se došlo do globalnog optimuma i prevazišli lokalni optimumi. Susjedstvo je okolina trenutnog rješenja x, a susjedstva se međusobno razlikuju po načinu na koji stvaraju nova rješenja. Proces lokalnog pretraživanja odvija se u iteracijama i zaustavlja kada u okolini novog rješenja ne postoji bolje rješenje, što je i osnovni nedostatak lokalnog pretraživanja. Prednost metode promjenljivih okolina je u tome što omogućava novu strategiju lokalnog pretraživanja, tako da raznim tehnikama prevazilazi zaglavljivanje u lokalnim minimumima.

Implementirana je i metoda promenljivih okolina, pri čemu je važno bilo odabrati pravo kodiranje i osmisliti koncept okolina koji je korišćen. Generisana inicijalna rješenja su bila dobar preduslov za primjenu VNS metode, jer su se značajno poboljšali lokalni optimumi, na testiranim instancama. Kao nastavak ovog istraživanja nastojaće se optimizovat istom metodom problem podjele u klastere u odnosu na dobijene rezultate [12] prezentovane u Tabeli III. Tako da se ovaj algoritam VNS metode uporedi sa drugim testnim rezultatima [12] i istaknu prednosti.

Metoda promjenljivih okolina ima brojne primjene u rješavanju raznih problema optimizacije. U mnogim životnim oblastima, istraživači se često sreću sa situacijama u kojima je potrebno objekte razvrstati u grupe homogenih objekata bilo da su individue, firme, proizvodi ili čak njihova ponašanja. Strateška rješenja bazirana na identifikaciji grupa pomažu u donošenju važnih odluka. Ova ista potreba se sreće u različitim područjima, od fizičkih do društvenih nauka.

Za proučavani problem, bilo je neophodno razmotriti i osmisliti efikasne i efektivne postupke smještanja objekata u klastere, neke društvene mreže. Efikasnost podrazumijeva dobijanje odgovarajuće podjele objekata u manje grupe (klastere) u što kraćem vremenu, a efektivnost se odnosi na kvalitet dobijenog rješenja sa što manjim brojem međuklasterskih veza. Međutim, postoji mnogo faktora koji utiču na efikasnost postupka smještanja osoba u klastere (broj osoba, broj klastera, maksimalan i minimalan moguć broj osoba u klasterima, ukupno utrošeno vrijeme, …), od kojih mnogi zavise jedni od drugih. Doprinos rada je program koji rješava problem podjele u klastere, generiše početno rješenje i primjenjuje postupak lokalnog pretraživanja. Program je implementiran u okviru rješavanja problema podjele u klastere tako da se minimizira normalizovan presjek, kao jedan od kriterijuma klasterovanja grafa, u društvenim mrežama. Proučavan optimizacioni problem podjele u klastere je šire primjenljiv u praktične svrhe. Tako isti programski kod se može lako prilagoditi različitim primjenama u svim oblastima ljudskih aktivnosti (biologije, anatomije, hemije, fizike, lingvistike, ekonomije, računarstva, sporta, muzike, ...).

Proučavani problem, postupka podjele u klastere je bio složen i zahtijevao je izvođenje velikog broja eksperimenata. Intuitivnim postupcima smještanja osoba u klastere došlo se intuitivno do parametara, koji su minimizirali vrijednost funkcije Ncut. Dalja istraživanja će se kretati u pravcu primjene iste ili drugih heuristika i njihovih kombinacija, na istom proučavanom problemu, u cilju što efikasnijeg rješavanja i optimizacije proučavanog problema klasterovanja.

LITERATURA

[1] D. Aloise and P. Hansen, “Clustering,” in Handb. of Discrete and Combinatorial Mathematics, D. Shier, Ed. CRC Press, to appear.

[2] Z. Wu and R. Leahy, “An optimal graph theoretic approach todata clustering: Theory and its application to image segmentation,” IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 15, pp. 1101–1113, 1993.

[3] N. Mladenović and P. Hansen, “Variable neighborhood search,” Computers and Operations Research, vol. 24, pp. 1097–1100, 1997.

[4] J. Shi and J. Malik, “Motion segmentation and tracking using normalized cuts,” in Proc. of the 6th Int’l Conf. on Computer Vision, 1998, pp. 1154–1160.

[5] J. Shi and J. Malik, “Normalized cuts and image segmentation,” IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 22, pp. 888–905, 2000.M.

[6] P. Hansen and N. Mladenovi´c, “Variable neighborhood search: principles and applications,” European Journal of Operational Research, vol. 130, pp. 449–467, 2001.

[7] S. Yu and J. Shi, “Multiclass spectral clustering,” in Int’l Conf. Computer Vision, 2003.

[8] M. Newman and M. Girvan, “Finding and evaluating community structure in networks,” Phys. Rev. E, vol. 69, p. 026113, 2004.

[9] M. Lagrange, L. Martins, J. Murdoch, and G. Tzanetakis, “Normalized cuts for predominant melodic source separation,” IEEE Trans. on Audio, Speech, and Language Processing, vol. 16, pp. 278–290, 2004.

[10] I. Dhillon, Y. Guan, and B. Kulis, “Weighted graph cuts without eigenvectors: A multilevel approach normalized cuts and image segmentation,” IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 29, pp. 1944–1957, 2007.

[11] P. Hansen, N. Mladenovi´c, and J. P´erez, “Variable neighborhood search: methods and applications,” 4OR, vol. 6, pp. 319–360, 2008.

[12] P Hansen, M Ruiz, D Aloise, “A Variable Neighborhood Search Heuristic for Normalized Cut Clustering,”, - 2010 - gerad.ca

ABSTRACT

This paper describes the development of efficient and effective methods of partitioning the members of a social network, with special emphasis on solving MIN-CUT optimization problems on graphs, which uses Ncut as the objective function. To solve the problem, the proposed method is changeable environment. This work is implemented within the framework of solving the problem of division into clusters so that the normalized cross-section, as one of the criteria clustering graph, optimizes the objective function in social networks.

Graph Partitioning - The MIN-CUT Problem

Rava Filipović