Primena PSO algoritma za particionisanje grafova

Darko Čapko, *Student Member, IEEE*, Aleksandar Erdeljan, *Member, IEEE*, Imre Lendak, *Student Member, IEEE*

***Sadržaj* — Algoritmi za podelu grafova se uspešno primenjuju u različitim oblastima, kao što su: podela elektronskih komponenti na štampanim pločama, podela Web stranica po različitim kriterijumima, podela modela podataka u slučaju distribuiranih i paralelnih proračuna i sl. Podela grafa, razmatrana u ovom radu, svodi se na rešavanje NP optimizacionog problema u kome se čvorovi grafa grupišu u particije sa ciljem da se minimizuju veze između particija uz ograničenje da veličine particija budu balansirane. U radu je opisan postupak modelovanja i podele grafa *Particle Swarm Optimization* algoritmom (PSO). Dobijeni rezultati pokazuju da se algoritam može uspešno primeniti za podelu grafa manjih dimenzija i da daje suboptimalno rešenje blisko optimalnom uz razumno trajanje proračuna.**

***Ključne reči* — Particionisanja grafova, PSO (*Particle Swarm Optimization*) algoritam.**

I. Uvod

Algoritmi za podelu grafova imaju primenu u brojnim sistemima gde je potrebno izvršiti podelu na podsisteme da bi se optimizovao njihov rad. Primeri ovakve podele su:

- VLSI tehnologije – gde je neophodno izvršiti optimalnu podelu tranzistorskih komponenti u grupe (na nivou sistema, štampanih ploča, čipova) tako da povezanost između komponenti u okviru različitih grupa bude što manja;
- Web aplikacije – podela dokumenata u cilju optimalne klasifikacije dokumenata, pretprocesiranja za PageRank algoritam, i dr.;
- Veliki modeli podataka – zahtevaju skupe računare za procesiranje, pa se distribucijom podataka i obrade na više jeftinijih računara značajno smanjuje cena hardvera;
- Paralelni proračuni nad grupama podataka.

Generalno posmatrano, u svakoj od pomenutih oblasti postoje međusobno spregnuti elementi koje je potrebno razvrstati u grupe tako da u svakoj grupi budu elementi koji su povezani čvrstim vezama. U idealnom slučaju, gde postoje disjunktni podskupovi elemenata, problem podele je trivijalan, ali praktični problemi su složeniji jer se tipično nameće ograničenje na veličinu grupe i tada se kod svake moguće raspodele elemenata pojavljuju veze između grupa. Time se postupak podele postavlja kao optimizacioni problem gde je potrebno naći odgovarajuću podelu tako da veze među grupama budu minimalne i ujedno da su sve grupe približno (ili tačno) jednake veličine. Može se pokazati da se postupkom minimizacije veza između grupa dobijaju grupe (maksimalno) čvrsto povezanih elemenata.

Jedan od načina podele sistema je primena algoritama za podelu grafova. Problemi podele grafova spadaju u grupu NP teško rešivih problema kombinatorne optimizacije [1], pa se zbog složenosti problema najšešće koriste razne heurističke metode [2] - [5] za podelu grafova. Neke od njih ([2], [3]) daju vrlo dobre rezultate za grafove manjih veličina (do 100 čvorova), dok se metoda opisana u [4] posebno dobro pokazala na grafovima velikih dimenzija (više miliona čvorova).

U ovom radu je za problem podele grafa primenjen *Particle Swarm Optimization* (PSO) algoritam - evolutivni algoritam inspirisan ponašanjem skupina životinja (npr. jata ptica, riba i sl.) [6]. U njemu svaka jedinka predstavlja jedan od načina podele elemenata po grupama, tj. svaka jedinka nosi informaciju kojoj grupi pripada svaki od elemenata sistema. U razmatranim slučajevima pretpostavlja se da je broj grupa unapred zadat.

U drugom poglavlju su definisani osnovni pojmovi i način kreiranja grafa, a u trećem poglavlju je definisan kriterijum optimalnosti za podelu grafova. Osnovne karakteristike PSO algoritma su opisane u četvrtom poglavlju, dok su u petom poglavlju prikazani rezultati dobijeni primenom ovog algoritma.

II. Modelovanje sistema grafom

Razmatrani sistemi se opisuju neorijentisanim grafom $G=(V,E)$ sastavljenim od čvorova (V) i grana (E).

Čvorom grafa se predstavlja element sistema koji je potrebno svrstati u neku grupu ili podsistem (npr. tranzistorska komponenta, Web dokument, električni element, i sl.). U opštem slučaju, može se desiti da određeni skup elemenata uvek mora biti obrađivan istovremeno, pa se u takvim situacijama takav skup elemenata (ukoliko nije previše velik) predstavlja jednim čvorom čija je težina jednaka broju elemenata navedenog

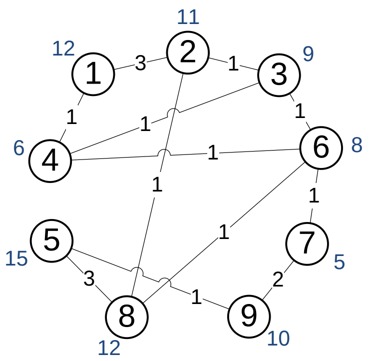
D. Čapko, Fakultet tehničkih nauka u Novom Sadu, Srbija (telefon: 381-21-459930; faks: 381-21-458873; e-mail: dcapko@uns.ac.rs).

A. Erdeljan, Fakultet tehničkih nauka u Novom Sadu, Srbija (telefon: 381-21-459930; faks: 381-21-458873; e-mail: ftn_erdeljan@uns.ac.rs).

I. Lendak, Fakultet tehničkih nauka u Novom Sadu, Srbija (telefon: 381-21-459930; faks: 381-21-458873; e-mail: lendak @uns.ac.rs).

skupa. Grane grafa predstavljaju veze između ovako definisanih čvorova. Pored toga, između dva čvora (elementa) može postojati više veza, pa se one predstavljaju jednom granom čija je težina jednaka ukupnom broju veza između čvorova (ili sumom težina tih veza).

Prema tome, problem podele grafa se bazira na podeli neorijentisanog grafa sa čvorovima i granama koji imaju određene težine, kao što je prikazano na slici 1.



Sl 1. Model sistema opisan grafom

Na slici 1. je prikazan graf sa 9 čvorova i 12 grana. Čvorovi su predstavljeni jedninstvenim identifikatorima (ID$\in${1,2,..,9}) i težinama (brojevi pored krugova), dok su grane predstavljene svojim težinama. Tako npr. čvor 1 sadrži 12 elemenata koji imaju 3 veze sa nekim od 11 elemenata iz čvora 2.

Značajni podaci za analizu i podelu grafa su vektor veličine čvorova *VC* i tzv. matrica povezanosti čvorova *PM*. U pitanju je kvadratna simetrična matrica, dimenzije jednake broju čvorova. Element matrice PM_{ij} predstavlja broj veza između čvorova *i* i *j*, i u proračunima se koristi samo gornja polovina matrice.

III. Definisanje kriterijuma optimalnosti

Potrebno je podeliti čvorove na unapred definisani broj grupa ili particija (*P*), tako da te particije budu približno istih veličina, a da njihova povezanost bude minimalna. Veličina particije *i* se definiše kao suma veličina svih čvorova koji u nju ulaze:

$$VP_i = \sum_{j \in P_i} VC_j \tag{1}$$

Particije treba da sadrže „dobro“ povezane čvorove (sa većim vrednostima elemenata u matrici *PM*) čija ukupna veličina ne prelazi maksimalno dozvoljenu veličinu (*MDV*):

$$MDV = (1+\varepsilon)\cdot\frac{1}{n_p}\sum_i VP_i \tag{2}$$

gde je:

VP_i – veličina particije *i*,

n_p – željeni broj particija,

ε – tolerancija odstupanja od srednje veličine particije.

Dozvoljene vrednosti za toleranciju su $\varepsilon \in [0, (n_p-1)/n_p]$. Ukoliko je $\varepsilon = 0$ tada su sve particije istih veličina (idealan slučaj balansiranosti), ali je i ograničenje za optimizacioni problem znatno striktnije, pa se do optimuma dolazi samo u specijalnim slučajevima grafova. Povećanjem tolerancije povećava se „manevarska“ sposobnost algoritma, pa se dobijaju bolje vrednosti kriterijumske funkcije.

Kriterijum optimalnosti je:

$$F = \max \sum_p f_p \tag{3}$$

gde je

$$f_p = \sum_{p,q \in P_k} PM_{p,q} \tag{4}$$

f_p - funkcija definisana za svaku particiju *k* (P_k), pri čemu su čvorovi *p* i *q* raspoređeni u P_k. Treba napomenuti da maksimum povezanosti unutar particije ujedno znači i minimum veza između particija.

U cilju balansiranosti veličina particija, potrebno je da bude zadovoljeno ograničenje:

$$VP_k \le MDV,\ \forall k \in \{1,2,..n_p\} \tag{5}$$

gde je VP_k veličina particije P_k određena u (1).

IV. PSO algoritam

PSO predstavlja stohastičku optimizacionu tehniku zasnovanu na kolektivnoj inteligenciji populacije. Populacija se sastoji od niza čestica koje se kreću kroz višedimenzioni prostor i menjaju svoje pozicije u zavisnosti od sopstvenog iskustva i iskustva ostalih jedinki [6].

Adaptacijom PSO algoritma na problem raspodele grafova, čestice koje predstavljaju neku raspodelu, imaju dimenziju jednaku broju čvorova koji se raspodeljuju. Pozicije čestica se ažuriraju u određenom broju iteracija u skladu sa definisanim kriterijumom optimalnosti, a traži se najbolja pozicija čestice.

Ažuriranje pozicije x_i čestice *i* započinje ažuriranjem njene brzine kretanja v_i za svaku dimenziju *d*:

$$\begin{aligned} v_{i,d} = {} & W\cdot v_{i,d} + C_1\cdot rand()\cdot\left(pBest_{i,d} - x_{i,d}\right) \\ & + C_2\cdot rand()\cdot\left(gBest_d - x_{i,d}\right) \end{aligned} \tag{6}$$

gde su:

W - faktor inercije,

C_1 - faktor individualnosti,

C_2 - socijalni faktor,

($pBest_{i,d} - x_{i,d}$) - individualna komponenta,

($gBest_d - x_{i,d}$) - socijalna komponenta,

rand() - slučajan broj u intervalu [0,1].

Velika brzina kretanja čestica rezultuje značajnom promenom pozicije, pa se uvodi ograničenje za brzine:

$$v_{i,d} = \begin{cases} -vSpan, & v_{i,d} < -vSpan \\ v_{i,d}, & -vSpan \le v_{i,d} \le vSpan \\ vSpan, & v_{i,d} > vSpan \end{cases} \tag{7}$$

Kako bi se obezbedila raznovrsnija brzine kretanja za sve dimenzije čestice, vrši se modifikacija ograničene brzine $v_{i,d}$, kao u formuli:

$$v_{i,d} = v_{i,d}\cdot\Delta x_{\max}\cdot rand() \tag{8}$$

gde je $\Delta x_{\max}\cdot rand()$ raspon za modifikaciju brzine.

PSO je kontinualan algoritam, stoga je potrebno prilagoditi ga razmatranom diskretnom problemu i definisati prostor u kome se čestice mogu pozicionirati. Prvo se vrši modifikacija brzine zaokruživanjem na najbliži ceo broj, kako bi dobili diskretnu vrednost:

$$v_{i,d} = round(v_{i,d}) \,, \tag{9}$$

a potom se ažuriraju pozicije čestica formulom:

$$x_i = x_i + v_i \,. \tag{10}$$

Pozicije čestica moraju imati celobrojne vrednosti iz intervala $[0,\, n_p)$, pa se na samom kraju radi modifikacija pozicija po modulu n_p:

$$x_{i,d} = \left|\mathrm{mod}\left(x_{i,d}, n_p\right)\right| \tag{11}$$

gde je $mod(a,b)$ ostatak pri deljenju a sa b.

Za ažuriranu poziciju x_i čestice i izračunava se vrednost kriterijuma optimalnosti $F(x_i)$. Pozicija x_i se pamti kao najbolja na nivou individue $pBest_i$ i populacije $gBest$, ukoliko je bolja od dotadašnje najbolje pozicije.

Nakon izvršavanja definisanog broja iteracija PSO algoritma dobija se raspodela čvorova po particijama koja je sadržana u najboljoj poziciji čestice na nivou populacije.

V. REZULTATI

Primena algoritma će biti opisana na primeru grafa sa slike 1. Za graf se definiše vektor veličine čvorova VC:

j	1	2	3	4	5	6	7	8	9
VC_j	12	11	9	6	15	8	5	12	10

i matrica povezanosti čvorova PM:

PM_{ij}		i								
		1	2	3	4	5	6	7	8	9
j	1		3	0	1	0	0	0	0	0
	2			1	0	0	0	0	1	0
	3				1	0	1	0	0	0
	4					0	1	0	0	0
	5						0	0	3	1
	6							1	1	0
	7								0	2
	8									0
	9									

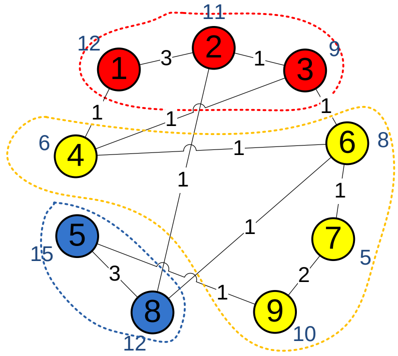
Neka je potrebno podeliti dati graf na 3 particije, tako da tolerancija odstupanja od srednje veličine particije bude 10% (n_p=3, ε=0.1). Na osnovu ovih parametara moguće je izračunati maksimalno dozvoljenu veličinu particije *MDV= 32.27.*

Jedinka x_i (za PSO algoritam) je vektor dužine jednake broju čvorova (u ovom primeru 9), gde vrednost j-tog elementa vektora predstavlja oznaku particije kojoj se čvor j pridružuje.

Primenom PSO algoritma na populaciji od 30 čestica nakon manje od 10 iteracija dobija se optimalna podela grafa:

$$x^* = \begin{bmatrix} 2 & 2 & 2 & 1 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}$$

što je prikazano na slici 2.



Sl.2. Prikaz grafa nakon podele na 3 particije

Urađeni su testovi i na grafu sa 54 čvora koji predstavlja obrađeni elektroenergetski model distributivne mreže grada Beograda gde su elementi mreže već grupisani u celine koje se napajaju iz istog izvora. Testovi (100 puta ponavljeni) su vršeni sa populacijom veličine 100 čestica za svaku od kombinacija vrednosti sledećih parametara:

- broj particija (n_p): 2, 3, 4, 5 i 6
- tolerancija ε: 0.1, 0.2 i 0.3
- iteracija PSO algoritma: 10, 100 1000 i 5000.

Imajući u vidu da su prethodno određene referentne vrednosti za podelu, utvrđena su odstupanja vrednosti funkcije optimalnosti dobijene PSO algoritmom od referentne optimalne vrednosti (što je prikazano u tabeli 1.). Referentna optimalna vrednost je dobijena dodatnim analizama modela elektroenergetske mreže i primenom algoritma *Brute Force* algoritma na povezane delove mreže. Ovaj postupak određivanja referentne optimalne vrednosti se može primeniti samo u specijalnim slučajevima grafova kod kojih je *PM* retka matrica, dok PSO algoritam nema takvih ograničenja.

Tabela 1. Relativna odstupanja optimalne vrednosti dobijene PSO algoritmom

n_p	ε	**10**	**100**	**1000**	**5000**
2	0.1	12%	2%	1.3%	1.3%
	0.2	10%	1.5%	0.5%	0.5%
	0.3	8%	1.5%	1.2%	1.0%
3	0.1	34%	17%	2.6%	2.3%
	0.2	28%	13%	1.8%	1.8%
	0.3	26%	13%	2.0%	2.0%
4	0.1	49%	30%	3.7%	3.2%
	0.2	41%	28%	1.3%	0.8%
	0.3	38%	25%	2.3%	2.3%
5	0.1	76%	53%	1.5%	0%
	0.2	59%	41%	3.2%	3.0%
	0.3	50%	34%	1.6%	1.6%
6	0.1	84%	79%	13%	13%
	0.2	72%	55%	2.2%	1.7%
	0.3	54%	39%	0%	0%

Na osnovu dobijenih rezultata može se zaključiti da PSO algoritam vrši dobru podelu grafa (sa relativnim odstupanjem od referentnog optimuma do 3.2%) za 1000 iteracija. U slučajevima kada je tolerancija za veličinu particije 10% (ε=0.1) i podelu na 6 particija (n_p=6), odstupanja od referentne vrednosti optimuma idu i do 13%. Prema tome, algoritam daje lošije rezultate u slučajevima kada su stroži optimizacioni kriterijumi – pronađe lokalni optimum i ostaje u njemu (posle 1000 iteracija nema nikakvog poboljšanja). Tada bi bilo poželjno korigovati pojedine parametre izraza (6) i nastaviti sa pretragom.

Pored toga, analizirana je balansiranost veličina dobijenih particija, tj. srednja vrednost odstupanja veličina grupa (O [%]) od izračunate srednje veličine particije (koja bi se dobila pri idealnoj balansiranosti). Navedena odstupanja su prikazana u tabeli 2.

Tabela 2. Opsezi dobijenih odstupanja veličina grupa od izračunate srednje vrednosti veličine

ε [%]	O [%]
10	4.9 - 6.9
20	9.8 - 13.8
30	14.6 - 22.8

Rezultati testiranja prikazani u tabeli 2. pokazuju da je PSO dao veoma dobre rezultate sa aspekta balansiranosti veličine grupa. Može se zaključiti da se grupe popunjavaju do ±70% dozvoljene tolerancije.

VI. Zaključak

U radu su opisan postupak modelovanja sistema grafom i primenjen je algoritam za podelu grafa na unapred zadati broj grupa. Ovaj algoritam daje dobre rezultate na grafovima manjih dimenzija, kako sa aspekta povezanosti elemenata, tako i u pogledu njihove balansiranosti po grupama (podjednake veličine grupa). PSO daje nešto lošije rezultate u slučajevima kada se traži veći broj particija i zadaju strožija ograničenja za veličinu particija. Opisani algoritam se može primeniti za particionisanje sistema u brojnim oblastima.

Literatura

[1] M.R. Garey, D.S. Johnson, *Computers and Interactability: A Guide to the Theory of NP-Completeness*, San Francisco, W.H.Freeman, 1979,

[2] B.W. Kerninghan, S. Lin, “An Efficient Heuristic Procedure for Partitioning Graphs”, *Bell System Technical Journal*, Vol. 49, Feb.1970., pp. 291-307.

[3] C. M. Fiduccia, R.M. Matteyses, “A linear time heuristic for improving network partitions” *In: Proceedings of the 19th Design Automaton Conference*, 1982, pp. 175-181

[4] G. Karypis, V. Kumar, „Multilevel algorithms for multi-constraint graph partitioning“, *ACM/IEEE Supercomputing Conference*, Orlando, 1998,

[5] A. Abou-Rjeili, G. Karypis „Multilevel Algorithms for Partitioning Power-Low Graphs“, *IEEE International Parallel & Distributed Processing Symposium (IPDPS)*, 2006.

[6] J. Kennedy, R. Eberhart, “Particle Swarm Optimization”, *Proc. IEEE Int’l. Conf. on Neural Networks*, Pert, Australia, 1995. pp. 1942-1948

Abstract

Graph partioning algorithms are successfully implemented in very different scientific areas: organizing electric components on printed circuit boards, grouping web pages by different criteria, model partitioning when performing distributed or parallel computation, etc. The graph partitioning algorithm discussed in this paper groups graph vertices, at the same time minimizing the connections between the groups and balancing the number of vertices per group. Although this problem is of NP complexity, this paper shows a how it can be solved with the Particle Swarm Optimization (PSO) algorithm. The results show that the PSO algorithm can be used for partitioning smaller graphs and achieving suboptimal solutions with relatively good performance.

PSO ALGORITHM FOR GRAPH PARTITIONING

Darko Čapko, Aleksandar Erdeljan, Imre Lendak