

Optimizacija topologije SDH transportne mreže

Zoran Čiča, Grozdan Petrović

Sadržaj — U ovom radu prezentovaće se problematika optimizacije SDH mrežne topologije sa više SDH prstenova. Takođe će biti predstavljen algoritam koji vrši optimizaciju SDH mrežne topologije sa više prstenova, kao i rezultati simulacije koji demonstriraju ispravan rad algoritma.

Ključne reči — genetski algoritam, optimizacija, SDH, topologija

I. UVOD

Moderne telekomunikacione mreže nude sve raznovrsnije servise sa ciljem da privuku što više korisnika i da se poveća saobraćaj u mreži i time poveća profit koji ostvaruje vlasnik mreže. Međutim, sa druge strane novi servisi imaju sve veće zahteve u pogledu kvaliteta koji mreža mora da im obezbedi, pre svega visoki protoci, kao i minimalna kašnjenja i varijacije kašnjenja. Iz tog razloga veoma je važno da se transportne mreže, koje imaju ulogu prenosa korisničkih paketa, takođe neprestano razvijaju i usavršavaju da bi mogle da prate pojavu novih servisa i njihove rastuće zahteve.

Zbog sve većeg saobraćaja koji transportne mreže moraju da prenesu za što kraće vreme očigledno su potrebne transportne mreže visokih kapaciteta pa su danas najznačajnije optičke transportne mreže koje ispunjavaju ovaj uslov, a njihov razvoj i povećavanje kapaciteta koje nude i dalje traje. SDH mreže predstavljaju veoma pouzdanu i dobro razvijenu optičku transportnu strukturu koja se najčešće koristi kao transportna infrastruktura za druge telekomunikacione mreže.

U ovom radu će se izložiti problematika nalaženja optimalne SDH topologije u kojoj ima više prstenova. Pri nalaženju optimalne SDH strukture nailazi se na više problema koji variraju u zavisnosti od usvojenih pretpostavki i time čineći problem lakšim ili znatno težim. Takođe će biti dat predlog algoritma za nalaženje optimalne strukture SDH topologije pri čemu će na primeru biti demonstrirani rezultati rada algoritma.

II. PROBLEM OPTIMIZACIJE SDH MREŽA

U zavisnosti od usvojenih pretpostavki svojstava SDH mreže problemi optimizacije u ovoj oblasti su veoma raznovrsni. Tako npr. saobraćaj u prstenu može biti unidirekcion ili bidirekcion čime problemi u optimizaciji postaju značajno različiti. Ako je saobraćaj unidirekcion onda opterećenje linkova je jednoznačno određeno

(postoje samo dva slučaja) u zavisnosti od saobraćaja koji generišu i primaju čvorovi u SDH prstenu i time dalja optimizacija biva znatno olakšana jer ne mora da se radi dodatna optimizacija kao u slučaju bidirekcionog saobraćaja gde u zavisnosti od izbora smerova saobraćaja za svaki komunikacioni par opterećenje pojedinih deonica SDH prstena može značajno da varira, pri čemu je najznačajniji maksimalno opterećenje na deonici koje može da utiče na SDH nivo prstena, pa je time neophodno optimizovati na prstenu razmenu saobraćaja između čvorova prstena da bi se pravilno operetile sve deonice i dobio SDH prsten što nižeg nivoa. U radu [2] je pokazano da je ovaj problem NP-kompleksan i dat je aproksimativan algoritam za rešavanje ovog problema u polinomijalnom vremenu. U radovima [3] i [4] su takođe predstavljeni algoritmi za rešavanje ovog problema. U radu [5] je ovom problemu priđeno sa malo drugačijim pristupom. Uzeto je da se saobraćaj jednog para može deliti između prstena tj. da može ići u oba smera. Autori tog rada su pokazali da ovako postavljen problem može da se reši u polinomijalnom vremenu i dali su i algoritam za rešavanje ovog problema.

Takođe, drugi optimizacioni problem koji se javlja pri posmatranju samo jednog SDH prstena je nalaženje optimalnog redosleda čvorova u prstenu da bi se potrošilo što manje optičkog kabla pri instalaciji prstena tj. prsten mora biti što kraći. Ovaj problem je u stvari problem trgovačkog putnika koji je NP-kompleksan.

Kako vidimo samo pri posmatranju jednog prstena možemo da naiđemo na različite optimizacione probleme od kojih su neki NP-kompleksni. Tako da problem optimizacije u koji je uključeno više SDH prstenova postaje veoma složen. Otuda je najčešći pristup u rešavanju ovakvih problema razbijanje glavnog optimizacionog problema na niz potproblema koji su jednostavniji za rešavanje. U zavisnosti od usvojenih pretpostavki, optimizacijom potproblema možemo da dobijemo i najoptimalnije rešenje glavnog problema, ali je najčešći slučaj da razbijanjem na potprobleme gubimo najbolju optimizaciju glavnog problema, pa dobijamo približno optimalno rešenje.

Postoji nekoliko radova koji se bave nalaženjem optimalne topologije SDH mreže sa više prstenova. U [7] se vrši optimizacija pretpostavljajući da postoji jedan glavni prsten koji povezuje sve ostale prstenove i time sav tranzitni saobraćaj (saobraćaj koji samo prolazi kroz prsten, a krajnji korisnici pripadaju drugim prstenovima) prolazi kroz glavni prsten čineći time problem optimizacije nešto jednostavnijim. Pri tome u [7] se pretpostavlja da su prstenovi međusobno povezani tako što jedan čvor pripada glavnom i još jednom prstenu pri

Z. Čiča, Elektrotehnički fakultet u Beogradu, Srbija (e-mail: cicasy@etf.bg.ac.yu).

G. Petrović, Elektrotehnički fakultet u Beogradu, Srbija (e-mail: gpetrovic@etf.bg.ac.yu).

čemu se za razmenu saobraćaja između prstenova koristi DCS (Digital Cross-connect). Drugi način za povezivanje prstenova jeste da se koristi link između dva čvora koji pripadaju prstenovima koji se povezuju, pri čemu link igra ulogu korisnika za prstenove tj. saobraćaj čvora iz jednog prstena koji ide ka čvoru drugog prstena sa kojim je povezan je praktično korisnička pritoka za taj drugi čvor i onda nema potrebe za korišćenjem DCS uređaja, koji su obično znatno skuplji od ADM (Add-Drop Multiplexer) uređaja.

Ukoliko bi svi prstenovi bili ravnopravni tj. ako nema glavnog prstena koji bi prenosio tranzitni saobraćaj problem postaje znatno složeniji jer sad bi svaki prsten mogao da nosi tranzitni saobraćaj i tako poveća zahteve u pogledu za kapacitetom prstena, pa je problem optimizacije još složeniji jer bi se moralo voditi računa i o optimalnoj povezanosti čvorova da dužina prstena bude što manja, ali i o saobraćajnim korisničkim zahtevima pri čemu organizacija čvorova u prstenove treba da bude takva da smanji tranzitni saobraćaj u što većoj meri. Takođe, dodatni problem u optimizaciji SDH mrežne topologije sa više prstenova je i broj samih prstenova u optimalnoj konfiguraciji, a taj parametar ne možemo unapred da odredimo tako da i on postaje dodatni parametar u procesu optimizacije čineći tako problem još kompleksnijim.

Kao što se iz izloženog može videti ova problematika je veoma raznovrsna i složena, pri čemu su neki problemi atraktivniji i primenjiviji u praktične svrhe (kao npr. problem raspodele saobraćaja u bidirekcionim konfiguracijama SDH prstenova). Mnogi problemi još imaju dosta prostora za usavršavanje njihovih rešenja, a pozitivna strana je što neki od navedenih problema imaju svoje ekvivalente u drugim naučnim oblastima tako da nalaženje kvalitetnih rešenja za probleme u ovoj oblasti mogu da nađu primenu i u drugim oblastima.

III. ALGORITAM ZA NALAŽENJE OPTIMALNE SDH MREŽE SA VIŠE PRSTENOVA

U ovom poglavlju će biti izložen algoritam za nalaženje optimalne SDH transportne mreže sa više prstenova. Smatraćemo da su prstenovi unidirekcionni, a da se povezivanje prstenova vrši preko jednog zajedničkog čvora uređajem DCS. Ulazni parametri su lokacije čvorova SDH mreže i saobraćajni zahtevi korisnika pri čemu smatramo da su saobraćajni zahtevi simetrični tj. saobraćaj od čvora i do čvora j je jednak saobraćaju od čvora j do čvora i . Unidirekcionni SDH prstenovi u kombinaciji sa pretpostavkom simetričnog saobraćaja olakšavaju problem optimizacije jer su svi linkovi SDH prstena jednako opterećeni čime izbegavamo problem optimizacije opterećenja linkova. Takođe, smatraćemo da imamo dva moguća SDH nivoa – 155Mb/s i 622Mb/s, kao i ograničenje u maksimalnom broju čvorova po prstenu. Kriterijumska funkcija Ψ koju smo koristili u procesu optimizacionog postupka je:

$$\Psi = \sum_{i=1}^{N_r} \sum_{j=1}^{N_c} x_{ij} D_i + \sum_{i=1}^{N_c} \sum_{j=i}^{N_c} l_{ij} d_{ij} E_{ij} \quad (1)$$

gde je

x_{ij} - binarna promenljiva koja ima vrednost 1 ukoliko čvor j pripada prstenu i , a u suprotnom ima vrednost 0.

D_i - koeficijent cene čvora u prstenu i (zavisi od kapaciteta prstena)

l_{ij} - binarna promenljiva koja određuje da li link između čvorova i i j postoji (vrednost 1) ili ne (vrednost 0).

d_{ij} - rastojanje između čvorova i i j

E_{ij} - koeficijent cene za link između čvorova i i j

N_c - ukupan broj čvorova

N_r - ukupan broj prstenova

Za proces optimizacije je upotrebljen genetski algoritam. Genetski algoritam je pogodan za optimizaciju nelinearnih problema i zasniva se na simulaciji evolucionih procesa. Osnovu genetskog algoritma čini populacija hromozoma, pri čemu su hromozomi sastavljeni od gena. U procesu evolucije dolazi do kombinovanja hromozoma putem ukrštanja čime se mešaju njihovi geni tj. razmenjuju svoje osobine. Od novodobijenih hromozoma formira se nova generacija populacije u koju ulaze hromozomi sa najboljim osobinama. Ovaj proces se ponavlja generacijama. Na ovaj način se simulira proces evolucije u kojem se stimuliše opstanak najboljih jedinki tj. onih sa najboljim osobinama.

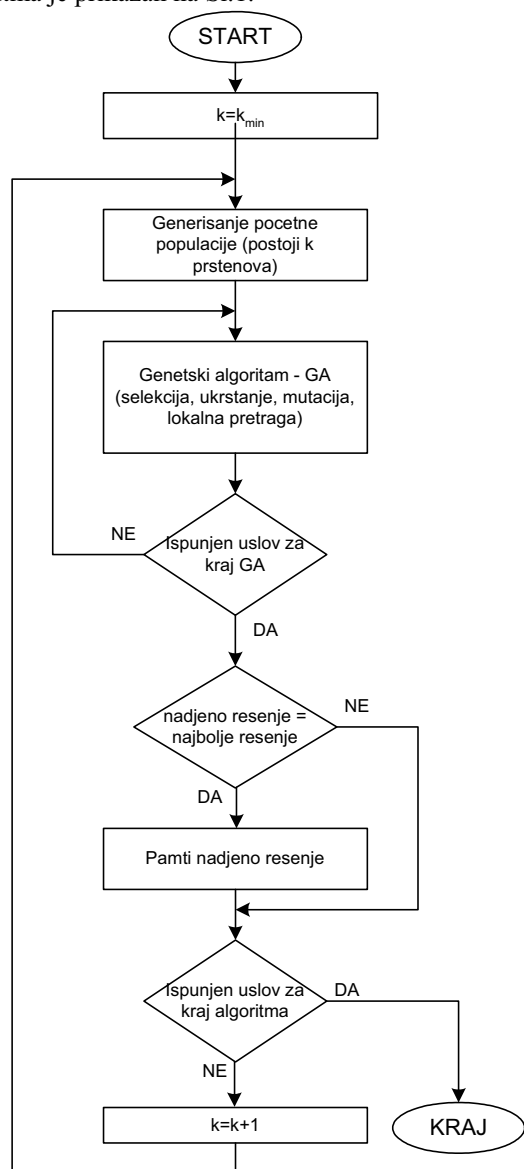
U našem slučaju gene hromozoma čine čvorovi sa podatkom kojim prstenovima oni pripadaju (pri tome je usvojeno da jedan čvor može da bude pripadnik maksimalno dva prstena). Pri generisanju početne populacije korišćena su dva pristupa. Prvi pristup je slučajno generisanje pripadnosti čvorova prstenu. Drugi pristup uzima u obzir međusobnu udaljenost čvorova, pa formira prstenove u skladu s tim parametrom i time pokušava da već u startu ostvari minimalne dužine prstenova. Oba pristupa se koriste za generisanje početne populacije tako da u početnoj populaciji imamo dve grupe hromozoma u zavisnosti kojim su pristupom dobijena.

Javlja se još dva problema – jedan problem je broj prstenova, a drugi je kako povezati prstenove u jedinstvenu celinu. Prvi problem smo prevazišli tako što pokrećemo genetski algoritam ponaosob za svaki broj prstenova počevši od minimalne moguće vrednosti (koja je određena pre svega maksimalnim brojem čvorova po prstenu, ali i saobraćajem) pa povećavajući taj broj za jedan sve dok u uzastopna dva ciklusa ne dođe do poboljšanja ili dok se ne dođe do maksimalnog broja prstenova (određeno uslovom da svaki nepovezan prsten mora da sadrži barem tri čvora) i tada se kompletan optimizacioni postupak završava. Za rešavanje drugog problema koristili smo princip slučajne pretrage za nalaženje najbolje kombinacije za povezivanje prstenova da bi izbegli dodatno komplikovanje optimizacionog algoritma. Važno je ovde napomenuti da gene čini pripadnost samo originalnom prstenu, dok pripadnost drugom prstenu (radi povezivanja prstenova u potpuno

povezanu celinu neki čvorovi pripadaju dvema prstenovima) se određuje već pomenutom slučajnom pretragom.

Dužinu rezultujućih povezanih prstenova minimizujemo koristeći jednu 'greedy' heuristiku za rešavanje problema trgovačkog putnika čime se dobija približno optimalna dužina za svaki prsten. Ova heuristika se zasniva na uključivanju redom najkraćih linkova od skupa čvorova koje treba da povežemo, pri čemu postoje dva uslova da se link ne uključi – prvi uslov je da se ne može u rešenje uključiti link ukoliko bi to bio treći link incidentan sa čvorom iz skupa, a drugi ukoliko bi taj link napravio petlju podskupa čvorova, a ne celog skupa čvorova za koji se rešava problem trgovačkog putnika. Ova heuristika se završava onda kada se broj uključenih linkova izjednači sa brojem čvorova u skupu.

Kao rezultat se dobija pripadnost čvorova prstenovima i na osnovu toga se lako nalazi saobraćaj na prstenovima i izračunava vrednost kriterijumske funkcije. Dijagram toka algoritma je prikazan na Sl.1.



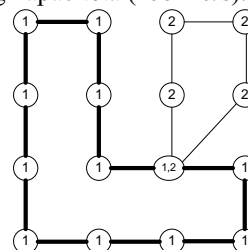
Sl. 1. Dijagram toka algoritma

Algoritam se izvršava tako što se fiksira broj prstenova u mreži na minimalnu vrednost. Za fiksnu vrednost broja

prstenova se generiše početna populacija hromozoma pri čemu se obezbeđuje da svakom prstenu pripadne bar tri čvora. Zatim se pokreće genetski algoritam u kome se u svakoj generaciji vrši selekcija i ukrstanje članova trenutne generacije čime se dobijaju potomci. Selekcija je turnir selekcija pri čemu je veličina turnira dva, a ukrstanje je jednopoziciono pri čemu se tačka ukrštanja bira na slučajan način. Nad dobijenim potomcima se vrši proces mutacije. Zatim se nad potomcima vrši proces pretrage u njihovoj okolini za eventualnim boljim rešenjem. U sledeću generaciju se prenose najbolji potomci, kao i najbolji hromozom prethodne generacije ukoliko je on bolji od svih dobijenih potomaka. Ovaj proces se ponavlja sve dok se ne ispuni jedan od dva postavljena uslova za kraj genetskog algoritma. Prvi uslov je maksimalan broj generacija, a drugi uslov je da u deset uzastopnih generacija ne dođe do poboljšanja rešenja. Nakon što se okonča genetski algoritam, upoređuje se najbolje rešenje sa prethodno nađenim i ukoliko je dobijeno rešenje bolje od prethodnog ono se pamti. Zatim se povećava broj prstenova za jedan i postupak se ponavlja. Algoritam se završava onog trenutka kad je dostignut maksimalan broj prstenova ili ukoliko se u dva uzastopna ciklusa nije našlo optimalnije rešenje. Za izvršavanje algoritma je razvijeno softversko rešenje koje izvršava algoritam u skladu sa datim opisom algoritma.

IV. REZULTATI SIMULACIJE

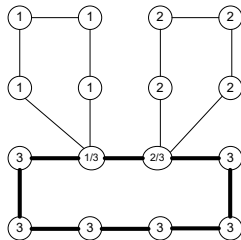
Za simulaciju je generisana mala mreža koja sadrži 16 čvorova, pri čemu svaki čvor komunicira sa svakim drugim čvorom u mreži saobraćajem od 2Mb/s po konekciji. Maksimalan broj čvorova po jednom prstenu je podešen na 15 čvorova. U genetskom algoritmu je uzeta veličina populacije četrdeset, a koeficijent mutacije 1%. Ovaj primer sa malim brojem čvorova je uzet da demonstrira ispravan rad algoritma jer je lak za praćenje i lako se utvrđuje da li je algoritam ispravan ili ne. U prvom slučaju stimulisaćemo koeficijentima kriterijumske funkcije rešenje sa prstenovima manjeg kapaciteta (koeficijenti SDH prstena nižeg kapaciteta su postavljeni na vrednost 1000 puta manju od koeficijenata SDH prstena većeg kapaciteta). Konačno dobijeno rešenje sadrži četiri prstena pri čemu sva četiri prstena su SDH prstenovi manjeg kapaciteta (155Mb/s).



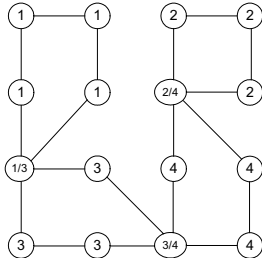
Sl. 2. Međurešenje sa dva prstena

Kretanje rešenja u zavisnosti od broja prstenova su prikazana na sl.2-sl.4 sa kojih se može videti kretanje optimalnog rešenja ka rešenju koje bi imalo samo prstenove manjeg kapaciteta. Deblje linije predstavljaju linkove SDH prstena većeg kapaciteta, a tanje linkove SDH prstena manjeg kapaciteta. Čvorovi nose broj prstena

come pripadaju.

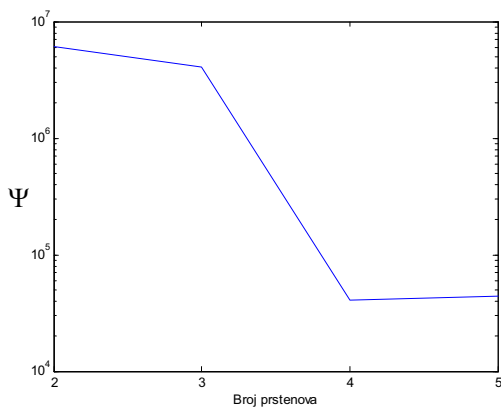


Sl.3. Međurešenje sa tri prstena



Sl.4. Krajnje rešenje sa četiri prstena

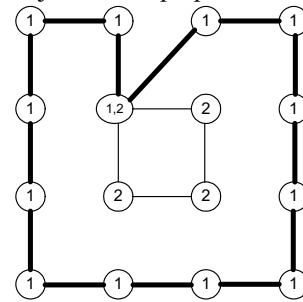
Na sl.5 je prikazano kretanje vrednosti kriterijumske funkcije u zavisnosti od broja prstenova, pri čemu se broj prstenova kreće od minimalne vrednosti dva do maksimalne vrednosti pet, a vrednost kriterijumske f-je je data u log razmeri. Kao što vidimo za vrednost broja prstenova pet dobijamo nepovoljniju vrednost kriterijumske funkcije, od optimalne koja je nađena za vrednost broja prstenova četiri. Takođe, može se primetiti veoma strm pad krive pri prelasku sa vrednosti broja prstenova tri na vrednost četiri, što se lako da objasniti činjenicom da se prešlo na rešenja koje sadrži samo prstenove manjeg kapaciteta. Otuda i malo povećanje kriterijumske funkcije za vrednost broja prstenova pet, jer se povećala ukupna dužina prstenova. U ovom slučaju tranzitni saobraćaj nije imao presudnu ulogu jer nije uticao na povećanje kapaciteta prstenova u nađenim rešenjima za broj prstenova četiri i pet, ali u opštem slučaju može da utiče na kapacitet prstenova u zavisnosti od povezanosti prstenova i pripadnosti čvorova prstenovima.



Sl.5. Zavisnost kriterijumske funkcije od broja prstenova

U drugom primeru se traži optimalno rešenje pri čemu je u ovom slučaju podešen isti koeficijent težine i za prsten manjeg i većeg kapaciteta. Za očekivati je da se dobije rešenje sa što manjim brojem prstenova jer je tako manje čvorova koji spajaju prstenove, a i manja je ukupna dužina linkova. Krajnje rešenje, predstavljeno na sl.6, sadrži ukupno dva prstena zbog ograničenja o

maksimalnom broju čvorova po prstenu.



Sl.6. Krajnje rešenje sa dva prstena

V. ZAKLJUČAK

Sledeći pravci unapređivanja prokazanog algoritma su poboljšanje rada pojedinih delova algoritma tako da se unapredi proces optimizacije (npr. TSP algoritam koji je primenjen ne daje uvek optimalne rezultate), ali je dobijena osnovna baza sa koje se dalje modularnim principom može unapređivati algoritam optimizacije, kao i uvoditi nove pretpostavke i time proširivati opseg problema koji bi razvijeno softversko rešenje pokrivalo.

ZAHVALNICA

Rad na ovom projektu je delimično finansiran sredstvima Ministarstva za nauku Srbije.

LITERATURA

- [1] "Routing, Flow, and Capacity Design in Communication and Computer Networks", Michal Pioro, Deepankar Medhi, Morgan Kaufmann Publishers
- [2] S.Cosares, I.Sainee "An optimization problem related to balancing loads on SONET rings" *Telecommunication Systems*, 3:165-181, 1994.
- [3] A.Schrijver, P.Seymour, P.Winkler "The ring loading problem" *SIAM Journal on Discrete Mathematics*, 11(1):11-14, February 1998.
- [4] S.Khanna "A polynomial time approximation scheme for the sonet ring loading problem" *Bell Labs Technical Journal*, 2(2):36-41, Spring 1997.
- [5] Y-S.Myung, H-G.Kim, D-W.Tcha "Optimal load balancing on SONET bidirectional rings" *Operations Research*, 45(1):148-152, January-February 1997.
- [6] M.Fischetti, J.J.Salazar Gonzalez, P.Toth "The symmetric generalised traveling salesman polytope" *Networks*, 26(2):113-123, 1995.
- [7] O.Goldschmidt, A.Laugier, E.Olinick, "SONET/SDH ring assignment with capacity constraints," *Technical report 01-EMIS-05*.
- [8] Z.Čiča, G.Petrović "Primena determinističke teorije servisnih sistema u planiranju transportnih mreža" *INFOTEH 2007*, Jahorina 2007.

ABSTRACT

This paper will present problems in SDH network topology with multiple rings optimization. The algorithm for optimization of SDH network topology with multiple rings will be presented, together with the simulation results.

SDH NETWORK TOPOLOGY OPTIMIZATION

Z.Čiča, G.Petrović