

DEFINISANJE SKUPA PODATAKA ZA VIZUELIZACIJU DISTRIBUTIVNE MREŽE**DEFINING THE SET OF DATA NEEDED TO VISUALIZE DISTRIBUTION NETWORK**

Sara Pavlović, Fakultet tehničkih nauka, Novi Sad

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – Savremenu distributivnu mrežu čini neizmeran broj entiteta, pa je njena sveobuhvatna vizuelizacija nepraktična za nadzor i upravljanje. Rješenje ovog problema je u prikazu samo važnih entiteta, tj. entiteta koji su od značaja korisniku, dok se preostali entiteti grupišu u cjeline. Da bi se postigao cilj, iskorišćena je radijalna struktura distributivne mreže. Mreža je opisana pomoću grafa i upotrebom Primovog algoritma kreirano je minimalno razapinjuće stablo koje obuhvata važne entitete mreže. Potom se stablo dijeli na odgovarajuće segmente, koji se šalju na prikaz. Rezultat ovakvog pristupa je jasna vizuelizacija distributivne mreže.

Ključne reči: Distributivna mreža, Vizuelizacija mreže, Minimalno razapinjuće stablo, Primov algoritam

Abstract – A modern distribution network contains an immense number of entities, so its overall visualization is impractical for supervision and control. Solution lies in the concept of showing only the most important entities of the network to a user: the ones that “carry” the most significant information. In order to achieve the goal, the radial structure of the distribution network is exploited. The distribution network is presented in a form of a graph, and by utilizing the Prim’s Algorithm, a minimum spanning tree which includes the important entities of the grid is obtained. Afterwards, the minimum spanning tree is partitioned into segments, which are further forwarded for displaying.

Keywords: Distribution network, Network visualisation, Minimum spanning tree, Prim’s Algorithm

1. UVOD

Dostignuća u polju informacionih i komunikacionih tehnologija su omogućila razvoj pametnih elektroenergetskih sistema [1]. Distributivni menadžment sistem (DMS) [2] je softverska komponenta u okviru pametnog elektroenergetskog sistema zadužena za manipulaciju podsistemom distribucije. Sa druge strane, distribucija elek. energije zasniva se na konceptu izvoda [3], elektroenergetskih veze koje napajaju dijelove potrošačkih područja.

S obzirom da savremenu distributivnu mrežu odlikuje milionski broj entiteta, njen sveobuhvatni prikaz je opterećen suvišnim detaljima, nepraktičan i neprikladan za nadzor i upravljanje. Vizuelizacija zasnovana samo na

entitetima koji su od interesa korisniku daje jasnu predstavu stanja mreže i olakšava njenu kontrolu. Cilj ovog rada je da definiše i objasni implementaciju algoritma kojim se određuje skup podataka za prikaz distributivne mreže.

2. MODEL DISTRIBUTIVNE MREŽE

Kako bi efikasno izvršavao zadatke DMS se oslanja na opis (model) mreže kojom upravlja. Takav model se najčešće zasniva na dobro provjerenom standardu koji propisuje strukturu modela. Najpopularniji standard je CIM (*Common Information Model*) [4].

Ovo istraživanje je bazirano na ideji da se oprema nebitna korisniku prikaže kao jedinstvena cjelina umjesto kao pojedinačni entiteti koji opterećuju vizuelizaciju distributivne mreže. Iz tog razloga od interesa su entiteti CIM-a koji opisuju provodnu opremu (*ConductingEquipment*), konektivnost između nje (*ConnectivityNode*), kao i grupisanje opreme u vidu kontejnera (*EquipmentContainer*). Takođe je iskoršćena mogućnost proširenja CIM-a za modelovanje izvoda distributivne mreže (*Feeder*), indikaciju koliko je oprema ili čvor mreže važan za korisnika (atribut *importance*), kao i grupisanje opreme koja se vizuelno prikazuje kao jedinstvena cjelina nazvana segment izvoda (*FeederSegment*).

3. MODELOVANJE DISTRIBUTIVNE MREŽE PREKO GRAFA

Rješenje problema koristi radijalnu prirodu distributivne mreže [3] kako bi uprostio detaljan model zasnovan na CIM-u. Entiteti i njihova konektivnost se predstavljaju grafom (tačnije strukturom tipa stablo) [5]. Nad novom organizacijom podataka primjenjuju se algoritmi za manipulaciju grafovima modifikovani u skladu sa potrebama ovog rada.

Prvi korak je modelovanje čvorova grafa. Čvorovi elektrodistributivne mreže, objekti tipa *Connectivity Node* u CIM-u, su ujedno i čvorovi grafa. Oprema u distributivnoj mreži (dvokrajnici, jednokrajnici i transformatori) služi za modelovanje grana grafa. Dodatno, svakoj grani i svakom čvoru je dodijeljena indikacija važnosti na osnovu *importance* atributa iz originalnog modela. Ovako predstavljena distributivna mreža omogućava jednostavnu identifikaciju važne opreme i važnih čvorova kao i putanja između njih.

2. INICIJALNO RJEŠENJE

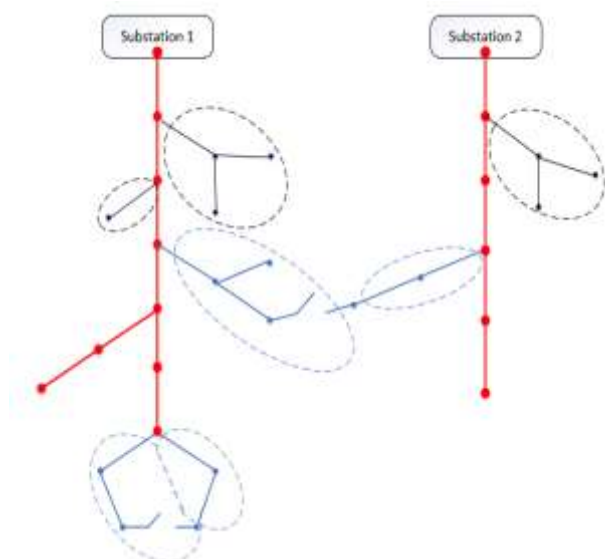
Segment izvoda definiše grupu opreme koja treba da se prikaže kao jedna cjelina. Podjela izvoda na segmente omogućuje prikaz samo onih elemenata koji su od značaja korisniku. Drugim riječima, ukoliko se korisnik odluči na

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Nemanja Nedić, docent.

prikaz preko segmenata dobija pojednostavljeni prikaz, što mu omogućuje fokusiranje na najvažnije entitete mreže.

Osnovni tipovi segmenta u ovom radu su definisani kao: *trunk*, *lateral* i *loop*. *Trunk* je glavna trofazna grana izvoda. Pored toga, segmentima tipa *trunk* su obuhvaćeni vodovi koji dosežu do entiteta (opreme ili *ConnectivityNode*-ova) proglašanih za važne od strane korisnika. *Lateral* je tip segmenta koji se nadovezuje na *trunk* i sadrži opremu bez petlje koja nije od važnosti za korisnika. U pojedinim slučajevima, distributivna mreža ima upetljani dio, radi obezbjeđivanja sigurnosti u napajanju potrošača iz više izvora ukoliko dođe do ispada dijela mreže usljed kvara. Takvi segmenti mreže čine segmente izvoda tipa *loop*. *Loop* segmenti mogu biti između dva povezana izvoda ili unutar jednog. Na slici 2.1 su prikazana dva susjedna izvoda. Crvenom bojom su označene glavne grane izvoda, kao i bočni vodovi koji su od interesa operateru. Crnom bojom su predstavljeni nevažni entiteti koji nisu uključeni u petlje, dok su plavom bojom označeni entiteti koji su dio petlje.



Slika 2.1. Grafički prikaz segmenata izvoda

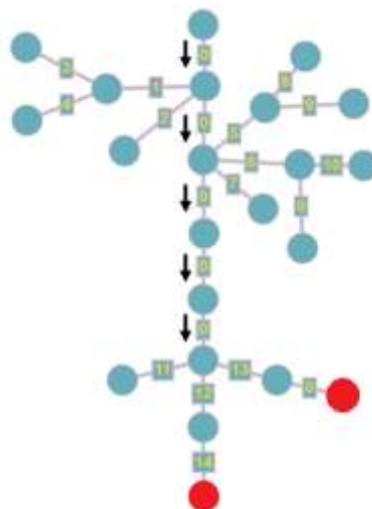
2.1. Detekcija povezanosti važnih entiteta

Svi važni entiteti, uz izuzetak električnih vodova, treba da se prikažu individualno. Važni električni vodovi koji se nadovezuju jedan na drugi mogu da se prikažu kao jedan segment. Iz tog razloga, prvenstveno je neophodno detektovati važne entitete i putanje između njih. U tu svrhu je prilagođen koncept minimalnog razapinjućeg stabla [5] koje će obuhvatiti sve važne grane i važne čvorove stabla.

S obzirom da graf kojim se modeluje izvod nije težinski, a Primov algoritam [6] za određivanje razapinjućeg stabla radi s težinskim grafovima, potrebno je dodijeliti težine granama grafa. Svim važnim granama se dodjeljuje težina 0. Za dodjeljivanje težina nevažnim granama je korišten *Breadth First Search* (BFS) algoritam [6].

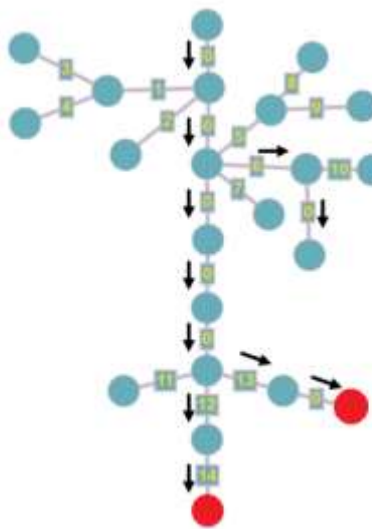
Nakon što je svakoj grani dodijeljena težina, slijedi formiranje "baze" minimalnog razapinjućeg stabla, koju čine samo važne grane, do kojih se može doprijeti krećući se Primovim algoritmom od početnog čvora. Na slici 2.2

su strelicama označene grane grafa koje čine bazu minimalnog razapinjućeg stabla.



Slika 2.2. Baza minimalnog razapinjućeg stabla

Važni čvorovi i važne grane koji nisu obuhvaćeni bazom minimalnog razapinjućeg stabla se nazivaju viseći. Neophodno je sve viseće čvorove i grane povezati sa bazom, kako bi se u cjelosti formiralo minimalno razapinjuće stablo. Za svaki viseći element pojedinačno se poziva Primov algoritam, gdje se kao početni čvor postavi baš taj viseći element. Obzirom da grane bliže početnom čvoru imaju manju težinu, Primov algoritam će se uvijek kretati prema bazi minimalnog razapinjućeg stabla. Algoritam se zaustavlja kada se dođe do čvora koji pripada minimalnom razapinjućem stablu. Na slici 2.3 je prikazan primjer grafa izvoda i minimalnog razapinjućeg stabla.



Slika 2.2. Graf izvoda i minimalno razapinjuće stablo

2.2. Region povezanih izvoda

Svaka promjena u izgrađenosti ili normalnom uklopnom stanju mreže utiče na jedan ili više izvoda, a time i na njihove segmente. Skup izvoda koji su direktno ili posredno povezani sa izvodima i koji su pod uticajem promjena, čine region povezanih izvoda.

S obzirom da se kreiraju segmenti izvoda čitavog regiona, neophodno je odrediti njegovo minimalno razapinjuće stablo. Prvenstveno se kreiraju razapinjuća stabla svakog izvoda posebno, a nakon toga se granični čvorovi koji pripadaju razapinjućem stablu izvoda, preko grana minimalne težine povezuju sa njegovim susjedom.

2.3. Kreiranje segmenata izvoda

Generisano minimalno razapinjuće stablo regiona predstavlja polaznu tačku za formiranje segmenata izvoda.

Segmenti izvoda tipa *trunk* se kreiraju od opreme koja pripada minimalnom razapinjućem stablu. Pretraga stabla ide u dubinu i obilaskom grafa se kreira jedan po jedan segment. Ispunjenjem jednog od sljedećih uslova, trenutni segment fidera se "prekida" i kreiraju se novi segmenti:

- Algoritam naišao na važan čvor.
- Algoritam naišao na granu koja modeluje važnu opremu koja nije sekcija.
- Algoritam naišao na čvor sa više od dvije susjedne grane, koje pripadaju minimalnom razapinjućem stablu.

Skup opreme, koja pripada segmentima tipa *trunk* se može prikazati kao jedna grana. Od ostatka opreme se kreiraju segmenti izvoda tipa *lateral* i *loop*.

U nastavku će biti opisan algoritam za kreiranje segmenata izvoda koji sadrže isključivo nevažnu opremu. *Loop*-ovi se javljaju u paru i grade upetljanu strukturu - krećući se granama petlje, od čvora preko kog je petlja zakačena na minimalno stablo, moguće je doći do istog čvora ili nekog drugog koji, takođe, pripada minimalnom stablu, istog ili različitog izvoda. Petlja unutar jednog izvoda povezuje čvorove jednog minimalnog stabla. Ako ovakva petlja sadrži normalno otvoren prekidač, onda se kreiraju dva segmenta tipa *loop*, razdvojena tim prekidačem. Petlja između dva izvoda povezuje čvorove minimalnih stabala tih izvoda.

Svi ostali segmenti, koji su izgrađeni samo od nevažne opreme su tipa *lateral*. Za svaki čvor minimalnog stabla, algoritam traži susjedne grane, koje ne pripadaju minimalnom stablu. Za svaku takvu početnu granu kreira novi segment fidera, čiji tip zavisi od uslova opisanih u prethodnim pasusima. U polje *equipmentOnThePath* se dodaje oprema koju modeluje početna grana i grane do kojih se može stići iz početne grane, a koje ne pripadaju minimalnom stablu.

2.3. Analiza inicijalnog rješenja

Inicijalno rešenje se zasniva na akcijama kreiranja segmenta izvoda nad čitavim regionom, koji može imati veliki broj izvoda i izvršavanje algoritma može biti nezadovoljavajućeg trajanja. Stoga se nameće pitanje na koji način da se izbjegne kreiranje segmenata cijelog regiona, ukoliko dođe do promjene modela, naročito ako su promjene nerelevantne za segmente izvoda.

3. UNAPREĐENJE ALGORITMA

Analiza inicijalnog rješenja je uvela ideju detekcije situacija kada promjena modela distributivne mreže ne zahtijeva generisanje segmenata izvoda čitavog regiona. Potencijalno unapređenje kojim bi se izbjeglo nepotrebno kreiranje segmenata izvoda bi umanjilo trajanje izvršavanja algoritma, a time i čitavu aktivnost izmjene

modela distributivne mreže. Da bi se ovakav napredak postigao neophodno je prepoznati promjene koje nisu od značaja za kreiranje segmenata, kao i promjene koje neće uticati na čitav region, već samo na jedan izvod ili jedan segment. Promjene su definisane kolekcijama operacija za dodavanje, ažuriranje i brisanje entiteta.

3.1. Uvođenje graničnih segmenata izvoda

U potrazi za boljim performansama DMS-a i realizaciji naprednog rješenja prvi korak je uvođenje novih tipova segmenata izvoda – graničnih segmenata (*boundary trunk*, *boundary lateral* i *boundary loop*). Segment je granični ako sadrži granični čvor, tj. čvor koji je povezan sa opremom nekog drugog izvoda.

Promjene negraničnih segmenata uglavnom ne utiču na susjedne izvode i ostatak regiona. S druge strane, promjene graničnih segmenata mogu izazvati "lančane" promjene segmenata u ostatku regiona, pa je u tom slučaju jednostavnije i brže kreirati segmente izvoda u regionu ispočetka.

Stoga je pogodno prilikom kreiranja segmenata prepoznati one koji su granični, kako bi se izvršio odgovarajući postupak u slučaju promjene segmenata.

3.2. Filtriranje zahtjeva za promjenom modela

Dalje unapređenje rješenja podrazumijeva filtriranje zahtjeva za promjenom modela. Svi dodati i obrisani entiteti direktno utiču na konektivnost mreže, a samim tim i na segmente izvoda. Stoga se operacije koji opisuju ove akcije automatski proglašavaju za relevantne. Međutim, samo podskup operacija koje definišu ažuriranje entiteta imaju uticaj na segmente izvoda:

- Promjena čvorova za koje je zakačena oprema.
- Promjena važnosti opreme.
- Promjena debljine električnog voda koji predstavlja glavnu trofaznu granu (*trunk*).
- Promjena debljine bilo kog električnog voda koja uzrokuje promjenu važnosti tog voda.

Nakon filtriranja relevantnih promjena slijedi definisanje segmenata na koje one utiču.

3.3. Obrada promjena relevantnih za segmente izvoda

U daljem procesu kreiranje segmenta izvoda (tokom inkrementalne promjene modela distributivne mreže) analiziraju se relevantne promjene modela. Rezultat analize predstavljaju sljedeće kolekcije:

- izvodi nad čijim regionima je neophodno kreirati segmente,
- izvodi za koje je pojedinačno potrebno kreirati segmente,
- oprema koja se dodaje u postojeći segment, bez potrebe za izvršavanjem algoritma.

Za sve relevantne operacije dodavanja, brisanja ili ažuriranja entiteta se provjerava da li je segment izvoda, kome entitet pripada, granični ili ne. Ukoliko segment nije granični, njemu odgovarajući izvod se markira za obradu, u suprotnom čitav region je markiran.

Ako se dodaje nevažna oprema, a segment je tipa *lateral* ili *loop* (bili oni granični ili ne) oprema se markira za ubacivanje u postojeći segment.

3.4. Kreiranje segmenata izvoda

Nakon pripreme podataka izvršava se kreiranje segmenta izvoda nad željenim dijelom mreže. Prvenstveno se formiraju segmenti za izvode u regionima markiranim za obradu. Za izvode markirane za pojedinačnu obradu se segmenti kreiraju nezavisno, bez formiranja regiona. S obzirom na lančani uticaj promjene segmenata izvoda neophodno je tekući (novonastali) skup graničnih segmenta uporediti sa predašnjim. Ukoliko je do promjene došlo (npr. pojavio se novi granični trunk) postoji mogućnost da je takva promjena izazvala promjene i u segmentima susjednih izvoda. U tom slučaju, da bi se očuvao korektan poredak segmenata, potrebno je pokrenuti kreiranje segmenata nad čitavim regionom.

S obzorom na karakteristike *lateral* i *loop* segmenata (graničnih ili ne), oprema, pod pretpostavkom da je nevažna, se može direktno uključiti. Na taj način je izbjegnuto pokretanje algoritma za kreiranje segmenata cijelog izvoda ili regiona izvoda. Kada se oprema direktno uključuje u postojeći segment neophodno je proširiti kolekciju koja definiše opremu datog segmenta identifikatorom opreme.

4. EKSPERIMENTALNA STUDIJA

Implementirana rješenja su primjenjena na regionu, koji sadrži 76 izvoda realne distributivne mreže. U Tabeli 4.1 prikazana su prosječna vremena izvršavanja algoritama kada se dodaje oprema u mrežu (scenario 1) i kada se briše oprema iz mreže (scenario 2).

Tabela 4.1. Vremena izvršavanja algoritama prilikom dodavanja i brisanja opreme u negraničnim segmentima

	Osnovni algoritam [ms]	Unaprijeđeni algoritam [ms]
Scenario 1	12246	33
Scenario 2	12569	205

U Tabeli 4.2 su prikazana prosječna vremena potrebna za dodavanje opreme koja povezuje 2 izvoda. Ovaj scenario uzrokuje kreiranje segmenata nad cijelim regionom izvoda kada je u pitanju osnovni i unaprijeđeni algoritam.

Tabela 4.2. Vremena izvršavanja algoritama prilikom dodavanja opreme u granične segmente

	Osnovni algoritam	Unaprijeđeni algoritam
Vrijeme [ms]	11896	11964

Na osnovu eksperimentalne studije zaključeno je sljedeće:

- Unaprijeđeno rješenje donosi značajna poboljšanja po pitanju performansi u odnosu na inicijalno rješenje u pojedinim slučajevima. Brzina izvršavanja je znatno veća kada se ažurira postojeći segment ili kreiraju segmenti jednog izvoda. Unapređenje naročito dolazi do izražaja kada region obuhvata veliki broj izvoda.
- U preostalim slučajevima, unaprijeđeno rješenje takođe podrazumijeva kreiranje segmenata nad svim izvodima u regionu, odnosno svodi se na inicijalno rješenje. Pri tome, dodatna obrada informacija u okviru unapređenog algoritma ne donosi primjetno usporjenje.

5. ZAKLJUČAK

Eksperimentalna studija je potvrdila efikasnost i korisnost implementiranih algoritama. Oba pristupa, inicijalni i unaprijeđeni, obezbjeđuju smanjenje nivoa detaljnosti prikaza distributivne mreže, tako da prikaz postaje jasniji i omogućuje fokus na najvažnije entitete mreže. Drugim riječima, rješenje zasnovano na segmentima izvoda je omogućilo prikaz samo onih entiteta koji su od značaja korisniku.

Dodatno analizom performansi, unaprijeđeni pristup je postigao više nego zadovoljavajuće rezultate. U pojedinim situacijama izmjene modela distributivne mreže, a samim tim i segmenata izvoda, čija frekvencija nije zanemarljiva, vrijeme izvršavanja unaprijeđenog algoritma je znatno kraće u odnosu na vrijeme izvršavanja inicijalnog rješenja. Međutim, očigledno je da postoje scenariji koji imaju uticaj na segmente čitavog regiona kada su performanse oba pristupa identične.

6. LITERATURA

- [1] R.DeBlasio, T.Cherry, "Standards for the Smart Grid", In Proceedings of International Conference IEEE Energy 2030, Atlanta, USA, pp. 1-7, 2008
- [2] D.Popović, "Power applications: A cherry on the top of the DMS cake", DA/DSM DistribuTECH Europe, Vienna, Austria, 2000.
- [3] V.Strezoski, "Osnovi elektroenergetike", Fakultet tehničkih nauka, Novi Sad, 2014.
- [4] "IEC 61970-301 (2009-04) Ed. 2.0 Energy management system application program interface (EMS-API)", 2009, ISBN 978-2-88910-593-9
- [5] R. Diestel, "Graph Theory", 4th Edition, Springer, ISBN 978-3-642-14278-9, 2012.
- [6] B. Stojanovic, "Algoritamske strategije - Materijali za predavanja", Prirodno matematički fakultet, Kragujevac

Kratka biografija:



Sara Pavlović rođena je u Vogošći 1995. god. Master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva, smjer Primjenjeno softversko inženjerstvo odbranila je 2018.god. Kontakt: srpvlvc@gmail.com