

# Višekriterijumska optimalna rekonfiguracija distributivne mreže primenom algoritma sivih vukova

Darko Šošić

Energetika/Katedra za elektroenergetske sisteme  
Univerzitet u Beogradu – Elektrotehnički fakultet  
Beograd, Srbija  
sosic@etf.rs

Predrag Stefanov

Energetika/Katedra za elektroenergetske sisteme  
Univerzitet u Beogradu – Elektrotehnički fakultet  
Beograd, Srbija  
stefanov@etf.rs

*Sažetak*—Rekonfiguracija distributivnih mreža predstavlja jednu mogućnost za poboljšanje karakteristika celog distributivnog sistema. Samu topologiju distributivne mreže je moguće menjati manuelno ili automatski, promenom stanja odgovarajućih prekidačkih elemenata, a sve u cilju smanjivanja gubitaka aktivne snage, povećanja sigurnosti sistema, ili poboljšanja kvaliteta električne energije. Rekonfiguracijom je moguće izvršiti i delimično balansiranje opterećenja pojedinih izvoda distributivne mreže. Promena uklopnog stanja distributivne mreže vrši se otvaranjem normalno zatvorenih (sekcionih) i zatvaranjem normalno otvorenih (poveznih) prekidača. Uslovi radikalnosti distributivne mreže i snabdevanje svih potrošača uvek moraju da budu zadovoljeni. Upotrebom jednokriterijumske optimizacije moguće je poboljšati samo jednu od karakteristika distributivne mreže, ali sa druge strane neka druge karakteristike mogu da se pogoršaju. Konfiguraciju distributivne mreže koja će da poboljšava više karakteristika celog sistema moguće je naći primenom višekriterijumske optimizacije. U ovom radu je, za tu svrhu, korišćen višekriterijumski algoritam sivih vukova. Predloženi algoritam rekonfiguracije je testiran na standardnoj simetričnoj IEEE 33 test distributivnoj mreži.

*Gljučne riječi*—distributivna mreža; rekonfiguracija; proračun tokova snaga; višekriterijumska optimizacija.

## I. UVOD

Rekonfiguracija distributivne mreže predstavlja proces promene topologije pojedinih izvoda otvaranjem i zatvaranjem sekcionih i poveznih prekidača. Cilj rekonfiguracije je poboljšanje pojedinih karakteristika distributivne mreže. Razmatrana poboljšanja mogu da se odnose na kraći ili na duži period razmatranog vremena. Prvi rad koji se bavio rekonfiguracijom, u kome je i uočen potencijal za istraživanje ove oblasti, su napisali Merlin i Back [1]. Počev od ovog rada, otvorena je nova oblast u kojoj su se različiti autori trudili da naprave algoritam za pronalaženje radikalne strukture distributivne mreže koja će imati najmanje gubitke aktivne snage, ali za prihvatljivo vreme izvršenja algoritma.

Posmatrano sa matematičkog stanovišta problem rekonfiguracije distributivne mreže spada u klasu složenih kombinatornih, nediferencijabilnih optimizacionih problema. Jedna od važnijih stavki koja ovaj problem dodatno usložnjava predstavlja uslov održavanja radikalne strukture distributivne mreže. Iterativna struktura proračuna tokova snaga, ograničenja tokova snaga po pojedinim elementima distributivne mreže, kao i neophodnost iscrpne pretrage svih mogućih konfiguracija dodatno doprinose složenosti razmatranog problema.

Metode za rešavanje problema rekonfiguracije distributivne mreže se mogu grubo podeliti u tri kategorije: 1) heurističke, 2) meta-heurističke i 3) matematičke optimizacione metode. Četvrtu grupu bi mogle da sačinjavaju hibridne kombinacije pomenutih metoda. Najpopularnije su metode koje se baziraju na heurističkim principima, uglavnom zbog brzog generisanja konačnog rešenja. Heurističke tehnike su zapravo optimizacioni procesi za nalaženje rešenja upotrebom karakteristika sistema, koje mogu biti definisane pomoću odgovarajućih koeficijenata. Ovi koeficijenti mogu da budu jednostavni (dobijeni na osnovu slobodnog nahođenja ili iskustva autora) ili sa druge strane mogu da se zasnivaju na rešavanju složenih problema pomoću analize osetljivosti potpunog ili uprošćenog matematičkog modela [2-7]. Međutim, ove metode su veoma zavisne od polazne konfiguracije distributivne mreže [8] tako da nalaženje globalnog optimalnog rešenja nije uvek zagarantovano.

Početkom devedesetih godina počeli su da se pojavljuju radovi koji su za rešavanje problema rekonfiguracije koristili meta-heurističke algoritme [9-15], grčki prefiks „meta“ znači „viši nivo“. Meta-heuristički pristup je specifičan način pretrage, koji započinje iz neke početne tačke (ili iz skupa početnih tačaka), prostora rešenja zadatog problema i navođenja početne tačke pomoću nekih pravila u cilju izbegavanja lokalnih optimuma. Glavna razlika između različitih meta-heurističkih metoda je način pretrage prostora rešenja. Ove metode mogu da daju optimalno rešenje nezavisno od početne konfiguracije distributivne mreže, ali su vremenski zahtevne tako da se mogu upotrebiti samo kada nije neophodno brzo dobijanje rešenja. U cilju rekonfiguracije distributivne mreže radi smanjivanja gubitaka aktivne snage i/ili minimizacije prekidačkih operacija korišćene su različite metode: sekvencijalno kodirani genetski algoritam [9], harmony search algoritam [10], modifikovana honey bee mating optimizacija [11]. Matematičke metode optimizacije, inače poznate i kao determinističke metode, za rešavanje problema rekonfiguracije koriste metode linearnog programiranja [16], dinamičko programiranje [17], ...

Primenom jednokriterijumske optimizacije moguće je unaprediti samo određene (razmatrane) karakteristike distributivne mreže. Sa druge strane, primenom višekriterijumske optimizacije moguće je naći rešenje koje će predstavljati kompromis poboljšanja više karakteristika. Opravdano je očekivati da će kompromisno rešenje biti lošije u odnosu na slučaj kada je traženo samo jedno optimalno rešenje, ali ono predstavlja rešenje kod koga su sve karakteristike podjednako zastupljene što nije bio slučaj kod jednokriterijumske optimizacije. U ovom radu optimizacioni postupak je korišćen za smanjivanje

gubitaka aktivne snage, smanjivanje maksimalnog pada napona, i balansiranje opterećenje po granama posmatrane mreže. Za ovu svrhu korišćen je osnovni algoritam sivih vukova [18, 19], jednokriterijumska optimizacija, i modifikovana verzija višekriterijumske optimizacije sivih vukova [20]. Svi proračuni su testirani na standardnoj simetričnoj IEEE 33 test distributivnoj mreži.

Rad je organizovan na sledeći način: u drugom delu su prikazani osnovni pojmovi višekriterijumske optimizacije. Osnovni algoritam sivih vukova je ukratko ponovljen u trećem delu, dok je u četvrtom delu opisano proširenje osnovnog algoritma sivih vukova na višekriterijumsku optimizaciju. U petom delu su opisane osnovne komponente neophodne za sprovođenje optimalne rekonfiguracije, dok je u šestom delu prikazan sam algoritam optimalne rekonfiguracije distributivne mreže. Rezultati su prikazani u sedmom delu, dok je zaključak rada dat u osmom delu.

## II. VIŠEKRIERIJUMSKA OPTIMIZACIJA

Višekriterijumska optimizacija se odnosi na optimizaciju problema koji ima više od jedne kriterijumske funkcije. Bez gubljenja na generalnosti, ovaj problem se može definisati kao minimizacioni problem na sledeći način

$$\text{Minimizirati: } \mathbf{F}(\mathbf{x}) = \{f_1(\mathbf{x}), \dots, f_k(\mathbf{x})\}, \quad (1)$$

Prema ograničenjima:

$$g_i(\mathbf{x}) \geq 0, \quad i = 1, \dots, M, \quad (2)$$

$$h_i(\mathbf{x}) = 0, \quad i = 1, \dots, N, \quad (3)$$

$$L_i \leq x_i \leq U_i, \quad i = 1, \dots, D, \quad (4)$$

gde je  $D$  ukupan broj upravljačkih promenljivih,  $K$  ukupan broj kriterijumskih funkcija,  $M$  je broj ograničenja tipa nejednakosti, dok  $N$  predstavlja ukupan broj ograničenja tipa jednakosti. Sa  $g_i$  je obeleženo  $i$ -to ograničenje tipa nejednakosti, a sa  $h_i$  je označeno  $i$ -to ograničenje tipa jednakosti. Donja i gornja granica  $i$ -te upravljačke promenljive je obeležena sa  $L_i$  i  $U_i$ .

Kod jednokriterijumske optimizacije pojedinačna rešenja mogu veoma lako da se upoređuju. Kada se traži minimalna vrednost nekog razmatranog problema rešenje  $\mathbf{x}^*$  je bolje od rešenja  $\mathbf{x}$  ako i samo ako je  $f(\mathbf{x}^*) < f(\mathbf{x})$ . Međutim, kada se radi sa višekriterijumskim problemima nije moguće rešenja porediti relacijskim operatorima. Neka su dati vektori  $\mathbf{y} = \{f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_k(\mathbf{x})\}$  i  $\mathbf{y}^* = \{f_1(\mathbf{x}^*), f_2(\mathbf{x}^*), \dots, f_k(\mathbf{x}^*)\}$  koji predstavljaju tačke u prostoru kriterijumskih funkcija  $\mathbf{y} \in O \subset \mathfrak{R}^K$ . Svaka komponenta vektora  $\mathbf{y}$  je vrednost objektivne funkcije koja je izračunata na osnovu dodeljene tačke iz prostora pretrage,  $\mathbf{x} \in S \subset \mathfrak{R}^D$ . Vektor  $\mathbf{y}^*$  dominira nad vektorom  $\mathbf{y}$  ( $\mathbf{y}^* \square \mathbf{y}$ ) ako je  $\mathbf{y}^*$  parcijalno manji od  $\mathbf{y}$ , odnosno

$$\begin{aligned} \forall k \in \{1, \dots, K\}: f_k(\mathbf{y}^*) \leq f_k(\mathbf{y}) \wedge \\ \exists k \in \{1, \dots, K\}: f_k(\mathbf{y}^*) < f_k(\mathbf{y}) \end{aligned} \quad (5)$$

Ako neki od ova dva uslova nije ispunjen onda rešenje  $\mathbf{y}^*$  ne dominira nad rešenjem  $\mathbf{y}$ . U skladu sa ovom definicijom moguća su tri slučaja:

- rešenje  $\mathbf{y}^*$  dominira nad rešenjem  $\mathbf{y}$ ,
- rešenje  $\mathbf{y}$  dominira nad rešenjem  $\mathbf{y}^*$ ,
- rešenja  $\mathbf{y}^*$  i  $\mathbf{y}$  su ravnopravna.

Rešenje nad kojim ne dominira ni jedno drugo izvodljivo rešenje naziva se Pareto optimalno, ili strogo efikasno. Pareto

dominantnost se odnosi na vezu vektora u domenu kriterijumskih funkcija,  $O$ , a ne u domenu prostora pretrage,  $S$ .

Pareto front je hiper-površ u prostoru kriterijumskih funkcija,  $O$ , koju sačinjava skup strogo efikasnih rešenja. Kao takav, Pareto front predstavlja skup najboljih kompromisa koja nisu dominantna jedna u odnosu na druga, tj. ne postoji objektivna funkcija koja bi mogla da se unapredi, a da se pri tom neka druga ne naruši.

## III. ALGORITAM SIVIH VUKOVA

U ovom odeljku će biti ukratko opisan jednokriterijumski optimizacioni algoritam koji se zasniva na ponašanju sivih vukova. Detaljan opis ovog algoritma je moguće naći u [18, 19]. Socijalni poredak i način lova sivih vukova su predstavljale osnovnu inspiraciju ovog algoritma. U cilju modelovanja hijerarhije u čoporu, tri najbolja rešenja su uzimana za  $(\alpha, \beta$  i  $\delta)$  vukove koji će donositi odluku, tj. upravljati tokom lova (potragom za najboljim rešenjem). Sva ostala rešenja predstavljaju  $\omega$  vukove. Nove lokacije  $\omega$  vukova definišu tri najbolja predstavnika čopora.

Da bi neki optimizacioni algoritam dobro radio neophodno je da poseduje dobar mehanizam pretrage. Grubo gledano pretraga se deli u dva segmenta: istraživanje i eksploataciju. U procesu istraživanja neophodno je generisati raznolika rešenja da bi se obuhvatio što širi prostor pretrage. Dok sa druge strane, kod procesa eksploatacije neophodno je pretraživati relativno uski deo oko najboljeg rešenja. Algoritam sivih vukova za ove potrebe koristi vektore  $\mathbf{A}$  i  $\mathbf{C}$  koji su definisani na sledeći način:

$$\mathbf{A} = 2 \cdot \mathbf{a} \cdot \mathbf{r}_1 - \mathbf{a},$$

$$\mathbf{C} = 2 \cdot \mathbf{r}_2. \quad (6)$$

Gde članovi vektora  $\mathbf{a}$  linearno opadaju od 2 do 0 sa tokom iteracija, a  $\mathbf{r}_1$  i  $\mathbf{r}_2$  su proizvoljni vektori na intervalu  $[0, 1]$ .

Pošto pozicija plena (optimalnog rešenja) nije unapred poznata, potraga za njim će se vršiti na osnovu informacija koje poseduju tri najbolja vuka. Svi  $\omega$  vukovi će unaprediti svoju poziciju na osnovu sledećih izraza:

$$\mathbf{D}_i = |\mathbf{C}_i \cdot \mathbf{X}_i - \mathbf{X}|, \quad i = 1, 2, 3. \quad (7)$$

$$\mathbf{X}(t+1) = \sum_{i=1}^3 (\mathbf{X}_i - \mathbf{A}_i \cdot \mathbf{D}_i) / 3. \quad (8)$$

Proces istraživanja je zagarantovan kada je  $|\mathbf{A}| > 1$ , i tada se vukovi udaljavaju od plena. Druga komponenta koja favorizuje istraživanje je vektor  $\mathbf{C}$ . Pošto je  $\mathbf{C}$  proizvoljan vektor u opsegu  $[0, 2]$  vukovi će se udaljavati od plena kada  $\mathbf{C} > 1$ , i približavati se plenu kada je  $\mathbf{C} < 1$ . Ovde je važno napomenuti da vektor  $\mathbf{C}$  može da uzme bilo koju vrednost iz pomenutog opsega u bilo kom trenutku optimizacionog algoritma. Drugim rečima, u krajnjim iteracijama algoritma može se desiti da neki vuk ode u istraživanje, upravo zbog vektora  $\mathbf{C}$ , a ne nastavi eksploataciju iako se približava kraj proračuna. Ova osobina pomaže optimizacionom algoritmu da izbegne lokalne optimume.

Proces eksploatacije je zagarantovan kada je  $|\mathbf{A}| < 1$ . Tada će se nova pozicija  $\omega$  vuka naći između njegove trenutne pozicije i pozicije plena koja je procenjena na osnovu informacija koje imaju vođe čopora.

Kao i većina drugih meta-heurističkih optimizacionih algoritama i ovaj algoritam započinje rad kreiranjem nasumično

odabrane početne populacije. Na svakoj iteraciji proračuna biraju se tri najbolja rešenja koja će predvoditi pretragu prostora rešenja. Pomeranje  $\omega$  vukova u prostoru pretrage se vrši pomoću izraza (6)–(8). Sa tokom iteracija vektori  $\mathbf{a}$  i  $\mathbf{A}$  se smanjuju, tako da  $\omega$  vukovi teže ka rešenju kada je  $|\mathbf{A}| < 1$  i udaljavaju se od rešenja kada je  $|\mathbf{A}| > 1$ . Na kraju optimizacionog postupka poslednji  $\alpha$  vuk predstavlja optimalno rešenje.

#### IV. VIŠEKRITERIJUMSKI ALGORITAM SIVIH VUKOVA

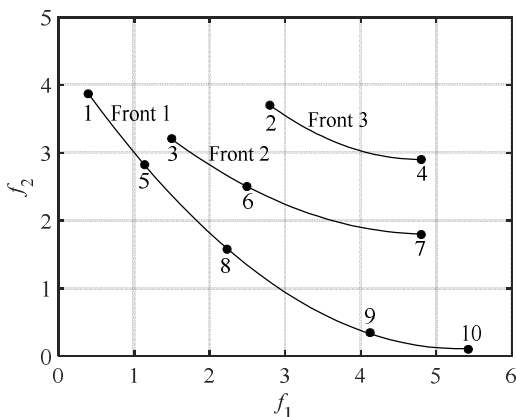
U cilju sprovođenja višekriterijumske optimizacije u osnovni algoritam sivih vukova potrebno je ugraditi još dve komponente. Jednu od ovih komponenta predstavlja arhiva u koju se smeštaju najbolja (nedominantna) rešenja. Druga komponenta se odnosi na način izbora vođa čopora ( $\alpha$ ,  $\beta$  i  $\delta$  vuka) iz arhive.

Arhiva predstavlja najobičniji registar u koji se smeštaju, i brišu, nedominantna rešenja koja su pronađena u dosadašnjem procesu pretrage. Jedna od bitnih stavki vezana za arhivu je i kontrola ulaska novih vektora u arhivu. Treba biti svestan i činjenice da arhiva ima unapred definisani broj mesta, tako da nije moguće u arhivu upisati više vektora od maksimalno dozvoljenog broja.

Da bi se umanjio broj upoređivanja uveden je koncept frontova iz NSGA-II algoritma [21]. Ona rešenja nad kojima ne dominira nijedan član u populaciji nazivaju se nedominantna rešenja prvog nivoa. Za ova rešenja se može reći da leže na najboljem frontu, odnosno na frontu nad kojim ne dominiraju preostali članovi populacije. Na Sl. 1 je prikazan jedan minimizacioni problem kod koga su rešenja podeljena na frontove. Prvi front kod ovog primera je sačinjen od skupa vektora čiji su redni brojevi 1, 5, 8, 9 i 10. Ako se rešenja iz prvog dominantnog fronta za trenutak eliminišu i ako se na preostala rešenja primeni princip sortiranja prema dominaciji dobija se sledeći dominantni front, vektori pod rednim brojem 3, 6 i 7 sačinjavaju drugi front, Sl. 1. Postupak se nastavlja sve dok se ne obrade sva rešenja u populaciji.

Na samom početku proračuna posle kreiranja početne populacije potrebno je u arhivu smestiti prvi front početne populacije. Ako prvi front ima manje od tri člana potrebno je iz drugog fronta uzeti proizvoljno odabran odgovarajući broj vektora. Neophodno je da arhiva ima najmanje tri člana. Ovo je uslovljeno osnovnim algoritmom sivih vukova pošto se pretraga navodi sa tri najbolja rešenja.

Kao što je već napomenuto, da bi se izbeglo nepotrebno poređenje lošijih rešenja, sa vektorima koji su smešteni u arhivi



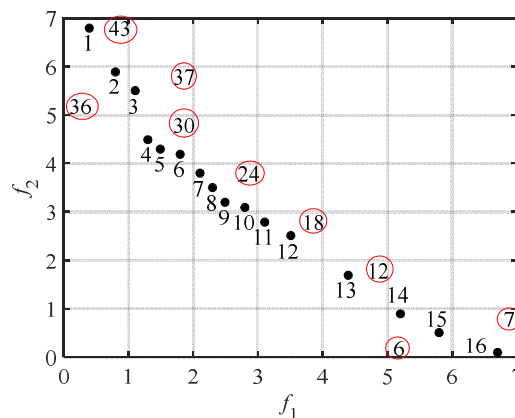
Slika 1. Kreiranje frontova.

će se porediti samo vektori iz prvog fronta trenutne populacije. Poređenje se odvija tako što se kompletna arhiva sjedini sa prvim frontom trenutne populacije, i ponovo se sprovede poređenje na principu dominacije. Novu arhivu će sačinjavati prvi front ovako formiranog skupa. U slučaju da je narušeno ograničenje broja članova arhive potrebno je odbaciti odgovarajući broj vektora, tj. popuniti arhivu do njenog maksimuma. Odluka o tome koji član arhive će se odbaciti donosi se na osnovu gustine broja rešenja u određenim delovima prostora kriterijumskih funkcija,  $O$ .

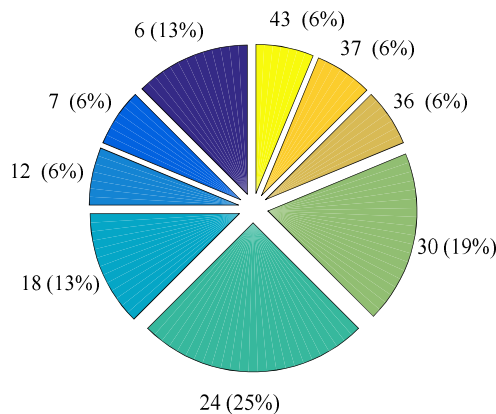
Domen kriterijumskih funkcija je moguće izdeliti na ravnomerne oblasti kao što je prikazano na Sl. 2. Nakon toga je potrebno pronaći i sačuvati okupirane oblasti (zaokruženi brojevi na Sl. 2) kao i broj vektora koji se u tim oblastima nalazi. Odluka o tome iz kog će se segmenta izbaciti vektor donosi se na osnovu pravila točka ruleta. Prvo je neophodno izdeliti oblasti na točku ruleta. Svako hiper oblasti se dodeljuje površina točka ruleta na osnovu sledećeg izraza:

$$p_i = n_i^b / \sum_{j=1}^m n_j^b \quad (9)$$

gde je  $p_i$  verovatnoća izbora  $i$ -tog segmenta,  $b$  je koeficijent pojačanja,  $n_i$  je broj vektora u  $i$ -toj hiper oblasti, dok je  $m$  ukupan broj hiper oblasti u kojima se nalazi najmanje jedan vektor. Na Sl. 3 je prikazana podele točka ruleta na segmente za primer sa Sl. 2. Zbog vidljivosti samog dijagrama koeficijentu pojačanja je dodeljena jedinična vrednost. Oblast koja ima najviše vektora, segment broj 24, ima 25% šanse da bude



Slika 2. Podela prostora kriterijumskih funkcija na hiper oblasti.



Slika 3. Točak ruleta.

odabrana. Ovaj procenat bi bio veći da je za  $b$  usvojena veća vrednost. Tako na primer, za  $b = 2$  ovaj segment bi zauzimao 42% površine točka ruleta, dok bi ta površina za  $b = 4$  iznosila 68%. Nakon podele se pomoću uniformne raspodele generiše broj iz opsega  $[0,1]$ , na osnovu koga se vrši odabir segmenta iz koga će se izbaciti jedan proizvoljno odabrani vektor. Ako je broj članova arhive i dalje veći od dozvoljenog potrebno je ponoviti opisani postupak sve do trenutka kada dimenzija arhive postane prihvatljiva.

Drugu komponentu predstavlja mehanizam odabira vođa čopora. Kod jednokriterijumske optimizacije tri najbolja rešenja su vodila pretragu u nadi da će naći rešenje koje se nalazi blizu globalnog optimuma. Međutim, kod višekriterijumske optimizacije nije moguće odrediti koja su rešenja najbolja. Zbog toga je bilo neophodno osmisliti neki mehanizam koji će omogućiti izbor vođa čopora. I za ovu svrhu se upotrebljava pravilo točka ruleta. Potrebno je favorizovati istraživanje oblasti koje imaju manje vektora da bi se ravnomerno pokrio ceo Pareto front. Za ovu svrhu se može iskoristiti izraz (9), sa tom razlikom što se koeficijentu pojačanja  $b$  dodeljuje negativna vrednost. Iz segmenta koji je odabran pomoću točka ruleta proizvoljno se bira jedan vektor i privremeno se sklanja iz arhive. Isti postupak se ponavlja i za ostala dva vodeća člana čopora. Kada su odabrana sva tri ( $\alpha$ ,  $\beta$  i  $\delta$ ) vuka koja predvode pretragu potrebno je u arhivu vratiti obrisane vektore. Dalji proces unapređenja  $\omega$  vuka se obavlja upotrebom izraza (6)-(8). Zbog činjenice da najbolja rešenja nije moguće odrediti kod rešavanja problema višekriterijumske optimizacije nove vođe čopora se biraju za svakog člana populacije, drugim rečima opisana procedura izbora vođa čopora će se ponavljati  $N_p$  (broj članova populacije) puta na svakoj iteraciji.

Sada je moguće ukratko prikazati višekriterijumski algoritam sivih vukova.

Kreirati početnu populaciju  $X_i$  ( $i = 1, \dots, N_p$ ).

Izračunati vrednosti kriterijumskih funkcija svakog vektora početne populacije.

Ispitati dominantnost i formirati frontove.

Popuniti arhivu sa prvim frontom i sa potrebnim brojem vektora iz viših frontova.

**while** ( $t <$  maksimalnog broja iteracija)

Izračunati vektore  $\mathbf{a}$ ,  $\mathbf{A}$  i  $\mathbf{C}$ .

**for** svaki vektor iz trenutne populacije

Izabrati  $\alpha$  vođu i privremeno ga izbaciti iz arhive.

Izabrati  $\beta$  vođu i privremeno ga izbaciti iz arhive.

Izabrati  $\delta$  vođu i privremeno ga izbaciti iz arhive.

Vratiti  $\alpha$ ,  $\beta$  i  $\delta$  vektore u arhivu.

Unaprediti pozicije trenutnog vektora pomoću izraza (6)-(8).

**end for**

Izračunati vrednosti kriterijumskih funkcija svakog novoformiranog vektora.

Ispitati dominantnost i formirati prvi front.

Sjediniti arhivu i prvi front trenutne populacije u pomoćni registar.

Ispitati dominantnost pomoćnog registra i formirati prvi front.

Smestiti prvi front u arhivu.

**if** veličina arhive veća od dozvoljene

Odrediti broj vektora,  $k$ , koje je potrebno izbaciti iz arhive

**for**  $i = 1$  do  $k$

Formirati mrežu i utvrditi broj vektora po hiper oblastima.

Pravilom točka ruleta odrediti iz koje će se oblasti izbaciti vektor.

Iz odabrane oblasti proizvoljno izabrati jedan vektor za izbacivanje.

**end for**

**end if**

$t = t + 1$

**end while**

## V. KOMPONENTE PRORAČUNA

U ovom poglavlju će biti opisane osnovne komponente koje se koriste u procesu optimalne rekonfiguracije.

### A. Upotreba osnovnih petlji

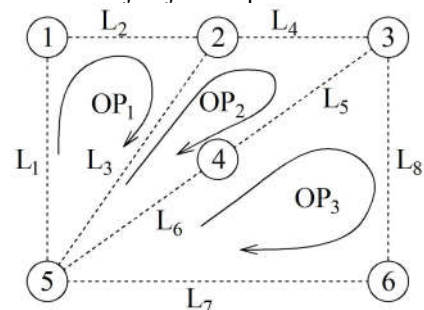
Izvodljiva konfiguracija distributivne mreže se formira upotrebom osnovnih petlji [22]. Kada se posmatra upetljana mreža broj osnovnih petlji,  $OP$ , se može izračunati pomoću izraza:

$$OP = N_{gr} - N_{cv} + 1 \quad (10)$$

Gde je  $N_{gr}$  ukupan broj grana u posmatranoj mreži, a  $N_{cv}$  je broj čvorova posmatrane mreže. Broj osnovnih petlji ujedno daje i informaciju koliko grana treba da bude otvoreno da bi mreža bila radikalna. Elemente jedne osnovne petlje sačinjava skup svih grana grafa koje formiraju posmatranu petlju. Grane osnovnih petlji za mrežu sa Sl. 4 su: Petlja 1 =  $[L_1, L_2, L_3]$ , Petlja 2 =  $[L_3, L_4, L_5, L_6]$ , Petlja 3 =  $[L_5, L_6, L_7, L_8]$ . Da bi se kreirala radikalna topologija potrebno je odabrati tačno jednu granu iz svake osnovne petlje. Međutim, kod ovog pristupa treba voditi računa da se ne izvrši izbor istih grane za različite osnovne petlje. Kao što se sa Sl. 4 može videti grana  $L_3$  pripada osnovnim petljama 1 i 2, dok grane  $L_5$  i  $L_6$  pripadaju osnovnim petljama 2 i 3. To znači da ako se odabere grana  $L_3$  u prvoj osnovnoj petlji, iz druge osnovne petlje je moguće izabrati vektore pod rednim brojem 2, 3 i 4, odnosno grane  $L_4, L_5$  i  $L_6$ .

### B. Provera povezanosti i radikalnosti

Kod nasumičnog odabira otvorenih grana u posmatranoj distributivnoj mreži veliki broj generisanih kombinacija će biti neizvodljiv, tj. formiraće se nepovezana ili upetljana mreža. Da bi se sprečio rad sa nedozvoljenim konfiguracijama distributivne mreže potrebno je izvršiti proveru povezanosti i radikalnosti. Provera radikalnosti se može iskoristiti i pre početka optimizacionog postupka da bi se odredile grane koje formiraju osnovne petlje. Algoritam za proveru radikalnosti je predložen u [23] i ovde će biti ukratko objašnjen. Za posmatranu mrežu potrebno



Slika 4. Primer upetljane mreže.

je formirati matricu povezanosti  $\mathbf{B}$ . Dimenzije ove matrice su  $N_{gr} \times N_{cv}$ . Nenulti elementi matrice  $\mathbf{B}$  su 1 i -1. Kada je grana  $i$  orjentisana od čvora  $j$  tada se na poziciju  $B(i,j)$  upisuje vrednost 1, u suprotnom ako grana  $i$  ulazi u čvor  $j$  onda se na poziciju  $B(i,j)$  upisuje -1. Kada se vrši određivanje grana osnovnih petlji prvo se upisuju grane stabla grafa a na kraju se dodaju samo jedna grana koja formira posmatranu petlju. Nakon formiranja matrice povezanosti,  $\mathbf{B}$ , potrebno je sabrati apsolutne vrednosti po kolonama ove matrice. Ako je suma apsolutnih vrednosti neke kolone jednaka broju 1 potrebno je pronaći tu vrstu i obrisati je iz matrice  $\mathbf{B}$ . Nakon toga se ponavlja opisani postupak sve do trenutka kada u sumi apsolutnih vrednosti kolona više nema broja 1. Grane koje su preostale predstavljaju grane petlje. U slučaju ispitivanja radikalnosti na kraju opisanog postupka matrica  $\mathbf{B}$  treba da ostane prazna. Opisani postupak se može lakše razumeti za primer određivanja grana prve osnovne petlje mreže sa Sl. 4 koji je prikazan u tabeli 1. Pošto na prvom koraku u redu sa sumama postoje dva broja 1, to znači da se dve grane moraju izbaciti iz matrice  $\mathbf{B}$ . Grane koje se izbacuju su obeležene isprekidanim pravougaonicima. Može se videti da nakon ove izmene i posle sabiranja apsolutnih vrednosti po kolonama, korak 2, postoji samo jedna kolona koja ima jedan element. Grana koja se izbacuje je grana broj 6, u tabeli 1 ova grana je obeležena punim pravougaonicima. Kada se izvrši sabiranje apsolutnih vrednosti po kolonama više neće biti kolona koje imaju samo po jedan nenulti element. Sve grane koje su ostale u matrici  $\mathbf{B}$  čine grane prve osnovne petlje. Ovim su samo potvrđuju rezultati koji se mogu videti sa Sl. 4, a koji su opisani u ranijem tekstu.

Povezanost je moguće proveriti pomoću determinante matrice  $\mathbf{B}$ . Pošto matrica povezanosti nije kvadratna potrebno je odstraniti kolonu koja odgovara referentnom čvoru i nakon toga izračunati njenu determinantu. Ako je vrednost determinante tako formirane matrice povezanosti jednaka 1 ili -1 onda je matrica radikalna i povezana u suprotnom ovaj uslov nije zadovoljen i takvo rešenje je potrebno odbaciti.

### C. Proračun tokova snaga

Proračun tokova snaga predstavlja jedan od osnovnih alata kod rešavanja problema rekonfiguracije. Za potrebe ovog rada dovoljno je koristiti najosnovniju varijantu proračuna tokova snaga [24] koja ne uvažava modifikacije koje nastaju zbog prisustva petlji i/ili distribuirane proizvodnje. Svaka iteracija se sastoji iz dva koraka, unapred i unazad. Kod koraka unazad računaju se gubici snage i snage na početku i kraju pojedinač-

TABELA I. ODREĐIVANJE GRANA PRVE OSNOVNE PETLJE

$\mathbf{B}$		Čvor					
		1	2	3	4	5	6
Grana	1	-1	0	0	0	1	0
	3	0	-1	0	0	1	0
	6	0	0	0	-1	0	0
Korak 1	2	1	-1	0	0	0	0
Korak 2	2	2	1	2	4	1	
Korak 3	2	2	0	0	2	0	

nih grana distributivne mreže. Ako se prijemni kraj grane obeleži sa  $r$ , a predajni sa  $s$  može se pisati:

$$P_s = P_r + R_{sr} \cdot (P_r^2 + Q_r^2) / |U_r|^2, \quad (11)$$

$$Q_s = Q_r + X_{sr} \cdot (P_r^2 + Q_r^2) / |U_r|^2. \quad (12)$$

Kod ovog koraka se kreće od krajnjih čvorova, tako da je potrebno snage potrošnje dodati odgovarajućim snagama na prijemnom kraju grane. Kada su poznate sve predajne snage moguće je u koraku unapred izračunati napone svih čvorova posmatrane mreže pomoću izraza:

$$\underline{U}_r = \underline{U}_s - (R_{sr} + jX_{sr}) \cdot (P_r^2 - jQ_r^2) / \underline{U}_s^*. \quad (13)$$

## VI. ALGORITAM

Sam algoritam optimalne rekonfiguracije u mnogome se poklapa sa opisanim algoritmom višekriterijumske optimizacije. Izračunavanja koja su potrebna za rekonfiguraciju distributivne mreže se izvršavaju u okviru koraka u kome se izračunavaju vrednosti kriterijumskih funkcija.

Prvo je potrebno izvršiti zaokruživanje upravljačkih promenljivih na najbliže cele brojeve. Nakon toga, potrebno je dekodirati upravljačke promenljive. Na primer, ako bi se za mrežu sa Sl. 4 generisao vektor  $\{1, 3, 4\}$  to bi značilo da je potrebno otvoriti grane:  $L_1$  (ovo je prva grana u prvoj osnovnoj petlji),  $L_5$  (ovo je treća grana u drugoj osnovnoj petlji) i  $L_8$  (ovo je četvrta grana u trećoj osnovnoj petlji). Sledeći korak predstavlja proveru povezanosti i radikalnosti generisane konfiguracije, ako ovaj uslov nije zadovoljen rešenje se momentalno odbacuje i generiše se novo. Nakon ove provere potrebno je pravilno orijentisati grane distributivne mreže i izvršiti proračun tokova snaga. Sa vrednostima koje se dobijaju na kraju proračuna tokova snaga moguće je izračunati vrednosti različitih kriterijumskih funkcija. U ovom radu će se razmatrati tri kriterijumske funkcije: gubici aktivne snage, maksimalni pad napona i izbalansiranost opterećenja pojedinačnih segmenata distributivne mreže. Ove kriterijumske funkcije se mogu prikazati u matematičkom obliku:

$$f_1 = \min \sum_{i=1}^{N_{gr}} R_{sr} \cdot (P_r^2 + Q_r^2) / |U_r|^2,$$

$$f_2 = \min \left( \max_{i \in [1, N_{cv}]} (|U_n| - |U_i|) \right), \quad (14)$$

$$Balans_i = (|I_{\max}| - |I_i|) / |I_{\max}|, \quad i = 1, \dots, N_{gr},$$

$$f_3 = \min \left( \text{median}_{i \in [1, N_{gr}]} (Balans_i) - Balans_i \right).$$

Gde  $I_{\max}$  predstavlja najveću vrednosti struje koja protiče kroz neki segment razmatrane distributivne mreže.

## VII. REZULTATI

Opisani algoritmi su testirani na standardnoj simetričnoj IEEE 33 test distributivnoj mreži [25]. Svi programi su samostalno napisani na i5 procesorskom računaru sa 4GB ram memorije.

U cilju poređenja rezultata sprovedeno je 7 različitih analiza. Prvo su pomoću jednokriterijumske optimizacije zasebno razmatrane pojedinačne kriterijumske funkcije (slučaj 1 –  $f_1$ , slučaj 2 –  $f_2$  i slučaj 3 –  $f_3$ ). U naredne tri analize je vršena višekriterijumska optimizacija različitih kombinacija dve kriterijumske funkcije (slučaj 4-  $f_1$  i  $f_2$ , slučaj 5 –  $f_1$  i  $f_3$ , i slučaj 6 –  $f_2$  i  $f_3$ ). Poslednji, sedmi slučaj razmatra istovremenu optimizaciju sve tri kriterijumske funkcije.

Broj upravljačkih promenljivih optimizacionog problema jednak je broju petlji u posmatranoj mreži, 5 u posmatranom slučaju. Moguće vrednosti pojedinačnih upravljačkih promenljivih su date u tabeli 2. Prva kolona teble 2 predstavlja redni broj, odnosno kod, grane u posmatranoj petlji. Ovim pristupom se olakšava primena optimizacionog algoritma, tako da su gornja ograničenja upravljačkih promenljivih definisana ukupnim brojem grana u posmatranoj osnovnoj petlji, odnosno {11, 10, 7, 15, 21}. Maksimalan broj iteracija za slučajeve 1 do 3 je 100, dok je za ostale slučajeve 1000. Broj članova populacije za sve proračune je bio isti i iznosio je 50, dok je koeficijentu pojačanja,  $b$ , dodeljena vrednost 4.

U tabeli 3 su prikazane vrednosti za sve slučajeve posle sprovedenog optimizacionog algoritma. Pored tražene kriterijumske funkcije, radi poređenja prikazane su i sve ostale vrednosti kriterijumskih funkcija. Za izdvajanje samo jednog rešenja iz celog Pareto fronta korišćena je metoda najmanje udaljenosti od utopija tačke. Zbog povezanosti prve i druge kriterijumske funkcije, slučaj 4, Pareto front se sastojao samo od dva rešenja. Takođe se može videti da se ova konfiguracija poklapa sa konfiguracijom koja je dobijena u slučaju broj 2.

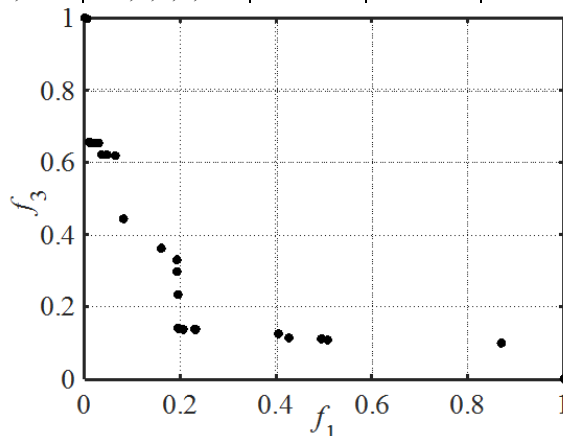
Pošto je za odabir rešenja sa Pareto fronta odabrano pravilo pomoću utopija tačke, za slučajeve 5, 6 i 7 dobijena je ista konfiguracija. Na Sl. 5-7 su prikazani Pareto frontovi za svaki slučaj posebno. Upotrebom različitih kombinacija težinskih faktora moguće je izvući neko drugo rešenje iz Pareto fronta. Pošto je smanjivanje gubitaka aktivne snage veoma značajnije od ostalih razmatranih kriterijuma, tako generisano rešenje bi težilo konfiguraciji u prvom slučaju.

TABELA II. MOGUĆE VREDNOSTI UPRAVLJAČKIH PROMENLJIVIH

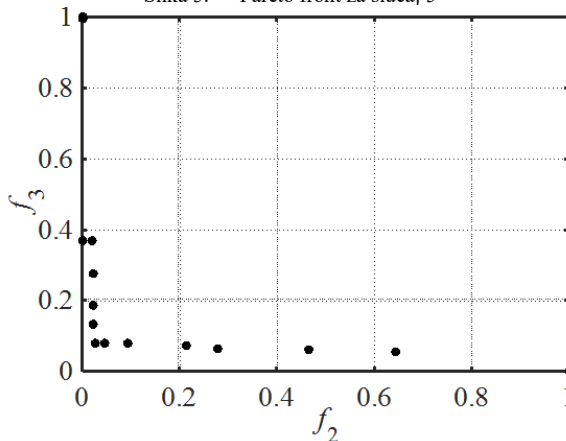
r.b.	OP1	OP2	OP3	OP4	OP5
1	3-4	2-3	9-10	2-3	6-7
2	4-5	3-4	10-11	3-4	7-8
3	5-6	4-5	11-12	4-5	8-9
4	3-23	5-6	12-13	5-6	9-10
5	23-24	6-7	13-14	6-7	10-11
6	24-25	7-8	14-15	7-8	11-12
7	6-26	2-19	9-15	8-9	12-13
8	26-27	19-20		9-10	13-14
9	27-28	20-21		10-11	14-15
10	28-29	8-21		11-12	15-16
11	25-29			2-19	16-17
12	19-20			19-20	17-18
13	20-21			20-21	6-26
14	21-22			21-22	26-27
15	12-22			12-22	27-28
16					28-29
17					29-30
18					30-31
19					31-32
20					32-33
21					18-33

TABELA III. REZULTATI ALGORITMA

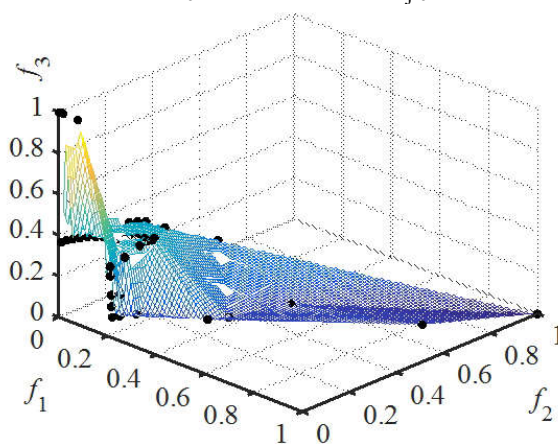
Slučaj	Kod grane	$f_1$	$f_2$	$f_3$
1	11, 6, 7, 10, 20	136.76	0.0621	0.4246
2 i 4	10, 6, 7, 10, 20	137.19	0.0587	0.5434
3	10, 4, 1, 14, 18	243.20	0.1342	0.2714
5, 6 i 7	10, 4, 7, 7, 21	157.61	0.0607	0.2929



Slika 5. Pareto front za slučaj 5



Slika 6. Pareto front za slučaj 6



Slika 7. Pareto front za slučaj 7

### VIII. ZAKLJUČAK

Rekonfiguracija distributivne mreže može da doprinese značajnim uštedama kroz smanjivanje gubitaka aktivne snage. Usled prisustva sve većeg broja merenja u distributivnoj mreži javlja se potreba za efikasnom metodom rekonfiguracije u realnom vremenu. U ovom radu je prikazana meta-heuristička



metoda za rešavanje problema rekonfiguracije izvoda distributivnih mreža. Ispitivane su tri kriterijumske funkcije: smanjivanje gubitaka aktivne snage, smanjivanje maksimalnog pada napona, i uravnoteženje opterećenja elemenata distributivne mreže. Pored jednokriterijumske optimizacije koja može da unapredi samo jednu karakteristiku distributivne mreže, a ostale da pogorša, u ovom radu je vršena i višekriterijumska optimizacija. Za ovu svrhu korišćena je modifikovana metoda sivih vukova. U slučajevima kada postoji veliki stepen korelacije između kriterijumskih funkcija nije moguće kreirati Pareto front, kao što je u radu i pokazano pri istovremenoj minimizaciji gubitaka aktivne snage i maksimalnih padova napona. Predloženi algoritam je testiran na standardnoj simetričnoj IEEE 33 test mreži.

#### ZAHVALNICA

Autori zahvaljuju Ministarstvu za nauku i tehnološki razvoj Republike Srbije koje je omogućilo izradu ovog rada u okviru Projekta III 42009 Inteligentne energetske mreže.

#### LITERATURA

- [1] A. Merlin and H. Back, "Search for a Minimal loss operating spanning tree configuration in an urban power distribution system," 5th Power System Computing Conference, Cambridge, UK, 1975.
- [2] S. K. Goswami and S. K. Basu, "A new algorithm for the reconfiguration of distribution feeders for loss minimization," IEEE Transactions on Power Delivery, vol. 7, pp. 1484-1491, July 1992.
- [3] D. Popović, D. Bekut, and V. Treskanica, Specijalizovani DMS algoritmi. Novi Sad: DMS Group, Jun 2004.
- [4] D. Shirmohammadi and H.W. Hong, "Reconfiguration of electric distribution networks for resistive line losses reduction," IEEE Transactions on Power Delivery, vol. 4, pp. 1492-1498, April 1989., " , "
- [5] W.-M. Lin and H.-C. Chin, "A New Approach for Distribution Feeder Reconfiguration for Loss Reduction and Service Restoration," IEEE Transactions on Power Delivery, vol. 13, pp. 870-875, July 1998.
- [6] F. Hosseinzadeh, B. Alinejad, and K. Pakfar, "A new technique in Distribution Network reconfiguration for loss reduction and optimum operation," presented at the 2009. CIRED 2009. 20th International Conference and Exhibition on Electricity Distribution, Prague, Czech Republic, 8-11 June 2009.
- [7] A. Bayat, "Uniform voltage distribution based constructive algorithm for optimal reconfiguration of electric distribution networks," Electric Power Systems Research, vol. 104, pp. 146–155, November 2013.
- [8] D. Sosic and P. Stefanov, "Uporedna analiza heurističkih metoda rekonfiguracije distributivnih mreža," presented at the 10. jubilarno savetovanje o elektrodistributivnim mrežama Srbije CIRED, Vrnjačka Banja, 29-30. Septembar 2016.
- [9] H.D.d.M. Braz and B.A.d. Souza, "Distribution Network Reconfiguration Using Genetic Algorithms With Sequential Encoding: Subtractive and Additive Approaches," IEEE Transactions on Power Systems, vol. 26, pp. 582-593, 2011.
- [10] R. S. Rao, K. Ravindra, K. Satish, and S. V. L. Narasimham, "Power Loss Minimization in Distribution System Using Network Reconfiguration in the Presence of Distributed Generation," IEEE Transactions on Power Systems, vol. 28, pp. 317–325, February 2013.
- [11] J. Olamaei, T. Niknam, and S. B. Arefi, "Distribution Feeder Reconfiguration for Loss Minimization Based on Modified Honey Bee Mating Optimization Algorithm," Energy Procedia, vol. 14, pp. 304-311, 2012.
- [12] E. M. Carreno, R. Romero, and A. Padilha-Feltrin, "An Efficient Codification to Solve Distribution Network Reconfiguration for Loss Reduction Problem," IEEE Transactions on Power Systems, vol. 23, pp. 1542-1551, 2008.
- [13] Q. Zhou, S. D., and L. W.-H.E., "Distribution feeder reconfiguration for operation cost reduction," IEEE Transactions on Power Systems, vol. 12, pp. 730-735, May 1997.
- [14] S. Bahadoorsingh, J. V. Milanović, Y. Zhang, C. P. Gupta, and J. Dragović, "Minimization of Voltage Sag Costs by Optimal

- Reconfiguration of Distribution Network Using Genetic Algorithms," IEEE Transactions on Power Delivery, vol. 22, pp. 2271–2278, October 2007.
- [15] D. P. Bernardon, V. J. Garcia, A. S. Q. Ferreira, and L. N. Canha, "Multicriteria Distribution Network Reconfiguration Considering Subtransmission Analysis," IEEE Transactions on Power Delivery, vol. 25, pp. 2684–2691, October 2010.
- [16] F. Llorens-Iborra, J. Riquelme-Santos, and E. Romero-Ramos, "Mixed-integer linear programming model for solving reconfiguration problems in large-scale distribution systems," Electric Power Systems Research 88 (2012) 137–145, vol. 88, pp. 137–145, July 2012.
- [17] M.-H. Shariatkah, M.-R. Haghifam, J. Salehi, and A. Moser, "Duration based reconfiguration of electric distribution networks using dynamic programming and harmony search algorithm," Electrical Power and Energy Systems, vol. 41, pp. 1–10, October 2012.
- [18] S. Mirjalili, S.M. Mirjalili and A. Lewis, "Grey Wolf Optimizer," Advances in Engineering Software, vol. 69, pp. 46-61, March 2014.
- [19] D. Sosic and I. Skokljek, "Primena algoritma Sivih Vukova na probleme u elektroenergetici," presented at the INFOTEH-JAHORINA, mart 2015.
- [20] S. Mirjalili, S. Saremi, S. M. Mirjalili, and L. d. S. Coelho, "Multi-objective grey wolf optimizer: A novel algorithm for multi-criterion optimization," Expert Systems With Applications, vol. 47, pp. 106-119,, April 2016.
- [21] K. Deb, Multi-Objective Optimization using Evolutionary Algorithms: John Wiley & Sons, 2001.
- [22] J. Mendoza, R. López, D. Morales, E. López, P. Dessante, and R. Moraga, "Minimal Loss Reconfiguration Using Genetic Algorithms With Restricted Population and Addressed Operators: Real Application," IEEE Transactions on Power Systems, vol. 21, pp. 948-954, May 2006.
- [23] T.T. Nguyen, A.V. Truong and T.A. Phung, "A novel method based on adaptive cuckoo search for optimal network reconfiguration and distributed generation allocation in distribution network," Electrical Power and Energy Systems, vol. 78, pp. 801–815, June 2016.
- [24] D. Rajicic, R. Ackovski and R. Taleski, "Voltage correction power flow," IEEE Transactions on Power Delivery, vol. 9, pp. 1056-1062, April 1994.
- [25] D. Šošić and V. Bečejac, "Rekonfiguracija distributivne mreže primenom metode jednostrukog zatvaranja petlji," presented at the INFOTEH 2016, Jahorina, Bosna i Hercegovina, 16-18 March 2016.

#### ABSTRACT

Reconfiguration of distribution networks is one option for improving the characteristics of the entire distribution system. Configuration of distribution system can be changed manually or automatically, by changing the status of the respective switching elements, with the aim of reducing power losses, increase system security, or improving the power quality. With reconfiguration is also possible to perform partial load balancing. The change in the network configuration is performed by opening normally closed (sectionalizing) and closing normally open (tie) switches of the network. These switchings are performed in such a way that the radiality of the network is maintained and all of the loads are energized. Using the single-criterion optimization is possible to improve one of the characteristics of the distribution network, but on the other hand some other characteristics can be worsen. The configuration of the distribution network that will improve the more characteristics of the whole system can be found using the multi-objective optimization. In this paper, for this purpose, multi-objective grey wolf optimizer was used. The proposed reconfiguration algorithm was tested on standard symmetrical IEEE 33 test distribution network.

#### MULTY-OBJEDTIVE OPTIMAL RECONFIGURATION OF DISTRIBUTION NETWORK USING GRAY WOLF OPTIMIZER

Darko Šošić, Predrag Stefanov