Optimiranje upravljanja preklopke distributivnog transformatora i snage distribuiranih izvora evolucijskim algoritmom

Mandarić, Dinko

Master's thesis / Diplomski rad

2016

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:200:135266

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-03-28**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

zir.nsk.hr

Image not found or type unknown

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU

FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA

Sveučilišni diplomski studij

OPTIMIRANJE UPRAVLJANJA PREKLOPKE DISTRIBUTIVNOG TRANSFORMATORA I SNAGE DISTRIBUIRANIH IZVORA EVOLUCIJSKIM ALGORITMOM

Diplomski rad

Dinko Mandarić

Osijek, 2016. godina

Sadržaj

1. UVOD 1

2. DISTRIBUIRANA PROIZVODNJA ELEKTRIČNE ENERGIJE 2

2.1. Definicija distribuirane proizvodnje 2

2.1.1. Svrha 2

2.1.2. Lokacija 2

2.1.3. Snaga 2

2.1.4. Područje isporuke el. energije 3

2.1.5. Tehnologija 3

2.2. Tehnologija proizvodnje električne energije 3

2.2.1. Energija sunca 3

2.2.2. Energija vjetra 4

2.2.3. Energija vode 5

2.2.4. Geotermalna energija 6

2.2.5. Biomasa 6

2.2.6. Gorive ćelije 7

3. TRANSFORMATORI S REGULACIJOM NAPONA 8

4. EVOLUCIJSKI ALGORITMI 11

4.1. Prirodna evolucija 11

4.1.1. Kromosomi i molekula DNK 11

4.1.2. Križanje i mutacija u prirodi 13

4.2. Vrste evolucijskih algoritama 15

4.2.1. Jedinka 16

4.2.2. Funkcija dobrote 16

4.2.3. Selekcija 16

4.2.4. Operatori varijacije 17

4.2.5. Algoritmi zasnovani na evoluciji 18

5. OPIS PROBLEMA OPTIMIRANJA UPRAVLJANJA PROIZVODNJOM DISTRIBUIRANIH IZVORA I PREKLOPKOM DISTRIBUCIJSKIH TRANSFORMATORA 19
5.1. Optimiranje općenito 19
5.2. Optimiranje upravljanja proizvodnjom distribuiranih izvora i preklopkom distribucijskog transformatora s ciljem smanjenja gubitaka 20
6. OPIS ALATA ZA SIMULACIJU 23
6.1. Populacija 24
6.2. Skaliranje funkcije cilja 25
6.3. Selekcija 26
6.4. Reprodukcija 27
6.5. Mutacija 27
6.6. Križanje 28
6.7. Migracija 28
6.8. Hibridna funkcija 29
6.9. Kriterij zaustavljanja algoritma 29
7. OBAVLJANJE SIMULACIJE 31
7.2. Parametri genetskog algoritma 34
7.2.1. Veličina populacije 34
7.2.2. Funkcija skaliranja 35
7.2.3. Funkcija selekcije 37
7.2.4. Križanje 39
7.2.5. Broj elitnih jedinki 41
7.2.6. Udio križanja 43
7.3. Obavljanje simulacije s konačnim postavkama algoritma 44
8. ZAKLJUČAK 51
9. LITERATURA 52
SAŽETAK 54
SUMMARY 55

ŽIVOTOPIS 56

Prilozi 57

1. UVOD

U novije vrijeme, ponajviše zbog štetnog utjecaja na okoliš počinje se razmišljati o promjeni ustroja mreže. Dosadašnji centralizirani elektroenergetski sustav pokušava se promijeniti na način da se proizvodne jedinice počinju priključivati u distribucijsku mrežu. Međutim, iako ovaj način ima svoje prednosti, dolazi i do nekih problema, kao što su zaštita koje je napravljena za zaštitu u samo jednom smjeru, ne mogućnost centralnog kontroliranja izvora proizvodnje i slično. U nekim slučajevima, kada je proizvodnja takvih izvora prevelika, dolazi do većih gubitaka, nego kada je mreža bila centralizirana.

Cilj ovog rada je upravo kontrola proizvodnje tih, distribuiranih izvora, s ciljem smanjenja gubitaka u mreži. Za rješavanje ovog problema koristit će se evolucijski algoritam. Evolucijski algoritmi su metode optimiranja koje oponašaju prirodnu evoluciju. Iz skupine evolucijskih algoritama odabran je genetski algoritam. Genetski algoritam koristi se za rješavanje problema optimizacije, te je pogodan i za rješavanje problema u ovom radu. Korišten je gotovi alat za genetski algoritam, koji je dio MATLAB programskog paketa.

Rad se sastoji od šest poglavlja. U prvom i drugom poglavlju teorijski su opisani distribuirani izvori i transformatori s regulacijom napona. U slijedećem poglavlju dan je pregled evolucijskih algoritama, te opis pojedinih dijelova. U četvrtom poglavlju opisan je problem kojim se rad bavi, a nakon toga dan je pregled i opis gotovog alata za genetski algoritam. U zadnjem poglavlju obavljena je simulacija, u kojoj je za odabranu mrežu sa ugrađenim distribuiranim izvorima, pomoću genetskog algoritma, napravljena optimizacija rada mreže s ciljem smanjenja gubitaka.

2. DISTRIBUIRANA PROIZVODNJA ELEKTRIČNE ENERGIJE

Od početka razvoja elektroenergetskih sustava (EES), uvijek se koristila ideja u kojoj su veliki središnji generatori proizvodili električnu energiju te ju preko transformatora injektirali u prijenosnu mrežu EES-a, koja se koristila za prijenos električne energije. Nakon toga se snaga iz prijenosne mreže, preko distribucijskih transformatora usmjeravala kroz srednjenaponsku i niskonaponsku mrežu do krajnjih potrošača na niskom naponu. Međutim, u novije vrijeme se počinje povećavati zanimanje za priključenje proizvodnih jedinica na distribucijsku mrežu. Ovaj način priključenja proizvodnih jedinica naziva se Distribuirana proizvodnja električne energije (eng. Distributed Generation - DG).

2.1. Definicija distribuirane proizvodnje

Za definiranje distribuirane proizvodnje postoji više parametara, a u ovom radu će se raspraviti samo neki od njih, a to su [2]:

- Svrha
- Lokacija
- Snaga
- Područje isporuke el. energije
- Tehnologija

2.1.1. Svrha

Svrha distribuirane proizvodnje je opskrbljivanje potrošača aktivnom električnom energijom.

Prema ovome distribuirani izvori ne moraju pružati jalovu, već samo radnu snagu.

2.1.2. Lokacija

Iako se definicija lokacije distribuirane proizvodnje razlikuje od autora do autora, u većini slučajeva se uzima da su distribuirani izvori oni koji su spojeni direktno na distribucijski dio mreže.

2.1.3. Snaga

Snaga distribuiranih izvora također se razlikuje od autora do autora. To je zato što se definicija snage distribuiranih izvora razlikuje od države do države. U ovom će radu, naime, snaga distribuiranih izvora biti do 300 MW, i to tako da je podijeljena u četiri skupine:

- *Mikro* - Proizvodne jedinice snage od ~1 W do 5 kW
- *Male* - Proizvodne jedinice snage od 5 kW do 5 MW
- *Srednje* - Proizvodne jedinice snage od 5 MW do 50 MW
- *Velike* - Proizvodne jedinice snage od 50 MW do 300 MW

2.1.4. Područje isporuke el. energije

Za područje isporuke el. energije uzima se lokalna distribucijska mreža, odnosno pretpostavlja se da se sva energija proizvedena u distribuiranim izvorima potroši unutar dijela distribucijske mreže na koji je spojena. Međutim, u nekim slučajevima distribuirani izvori noću proizvode više energije nego što je potrebno u lokalnoj mreži, te se u tom slučaju energija izvozi u prijenosnu mrežu.

2.1.5. Tehnologija

Kada se govori o distribuiranoj proizvodnji, najčešće se misli na energiju proizvedenu iz obnovljivih izvora energije, no to nije nužno tako. Iako se u većini slučajeva energija dobiva iz obnovljivih izvora energiju, ponekad se koriste i neobnovljivi izvori energije.

2.2. Tehnologija proizvodnje električne energije

Tehnologija proizvodnje el. energije može se podijeliti na obnovljive i neobnovljive izvore energije. Obnovljivi izvori energije su zastupljeniji u distribucijskoj proizvodnji, no nije rijetko da se koriste i neobnovljivi izvori.

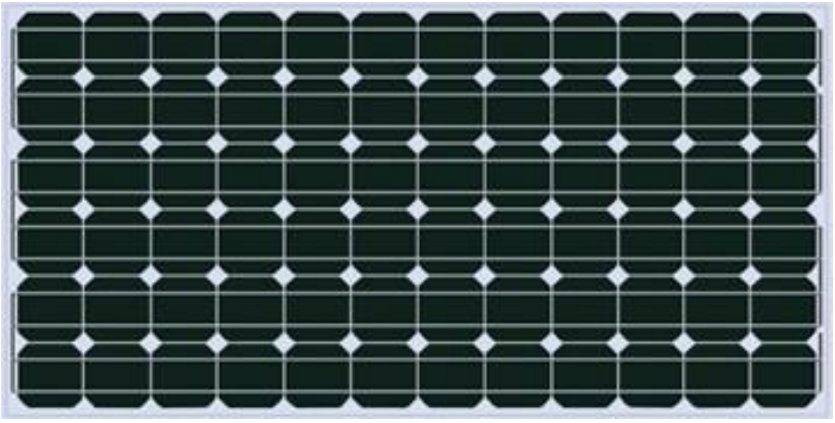
Pod obnovljive izvore energije ubrajamo one izvore energije koji se s vremenom ne mogu potrošiti, odnosno koji se s vremenom obnavljaju, kao što su energija sunca, vjetra, vode, geotermalna energija, energija iz biomase.

2.2.1. Energija sunca

Energija sunca se može koristiti na više načina, kao što su proizvodnja električne energije pomoću fotonaponskih ćelija, proizvodnja električne energije uporabom koncentrirane solarne energije, zagrijavanje vode ili zraka za kućanstva i slično. No za distribuiranu proizvodnju najčešće se koriste fotonaponske ćelije.

Fotonaponske ćelije koriste se za pretvorbu sunčeve energije u električnu. Sastoje se od 2 sloja, pozitivnog i negativnog, a razlika napona između ta dva sloja ovisi o intenzitetu sunčevog

zračenja. Sunčeva energija dolazi na zemlju u obliku fotona koji prilikom pada na fotonaponsku ćeliju predaju svoju energiju i na taj način "izbijaju" elektrone iz atoma, koji se nakon toga kreću prema drugoj (negativnoj) strani panela, te na taj način dolazi do razlike potencijala. Svaka ćelija daje struju od 2-4 A, ovisno o njenoj veličini, te napon od 0,5 V. Spajanjem ćelija u seriju postiže se odgovarajući napon. Fotonaponske ćelije spajaju se u jedinice koje se nazivaju fotonaponski paneli (Slika 2.1.) Nedostatak fotonaponskih ćelija je što imaju malu izlaznu snagu. Za veće snage zahtijevaju veliku površinu, i nisu primjenjive na svim područjima, jer im snaga proizvodnje ovisi o sunčevom zračenju.



Slika 2.1. *Fotonaponski panel sastavljen od 72 fotonaponske ćelije* [13]

2.2.2. Energija vjetra

Energija vjetra koristi se na način da prirodno strujanje zraka pokreće vjetroturbine. Vjetroturbine su energetski strojevi koji kinetičku energiju vjetra pretvaraju u mehaničku energiju koja se najčešće koristi za proizvodnju električne energije. Postoji više izvedbi vjetroturbina, ali se najčešće koriste vjetroturbine s vodoravnom osi, te trima lopaticama (Slika 2.2)

Vjetroturbina zajedno s električnim generatorom spojenim na osovinu vjetroturbine čini vjetroagregat. Protokom zraka, lopatice se počinju okretati, te okreću i osovinu na kojoj se nalaze, a time i rotor električnog generatora, te se na taj način proizvodi električna energija.

Najčešće se na jednoj lokaciji nalazi više vjetroturbina, koje se nazivaju vjetropark.

Nedostatak vjetroturbina je u tome što ih se može postaviti samo na mjestima na kojima ima dovoljno vjetra za njihov pogon, no to je uglavnom na slabo naseljenim mjestima, pa se takve proizvodne jedinice uglavnom spajaju na prijenosnu mrežu, a rjeđe na distribucijsku.



Slika 2.2. *Vjetroturbina s vodoravnom osi i trima lopaticama* [14]

2.2.3. Energija vode

Kada govorimo o energiji vode kod proizvodnje električne energije, najčešće se koristi kinetička energija vode u rijekama. U ovom slučaju su to mikro hidroelektrane, dok velike hidroelektrane imaju branu, te mogu proizvoditi puno više energije, te kontrolirati količinu proizvedene energije. Rjeđe se koriste i neki drugi oblici iskorištavanje energije vode kao što su kinetička energija valova, energija plime i oseke, snaga morskih struja i slično.

Kod distribuirane proizvodnje koriste se male hidroelektrane, koje se smatraju hidroelektranama snage do 10 MW.

Hidro elektrane koriste kinetičku energiju strujanja vode i u rotoru turbine ju pretvaraju u mehaničku energiju, koja se koristi za pokretanje rotora generatora, te se na taj način dobiva električna energija.

2.2.4. Geotermalna energija

Geotermalna energija je toplinska energija u zemljinoj kori nastala još kod formiranja zemlje prije 4 milijarde godina. Raspadanjem radioaktivnih elemenata, ta se toplina kontinuirano obnavlja te zbog toga geotermalnu energiju svrstavamo pod obnovljive izvore energije .

Za proizvodnju električne energije koriste se vruća voda i para. Koriste se tri osnovna tipa geotermalnih elektrana:

Princip suhe pare - koristi se iznimno vruća para (iznad 235˚C). Ta para koristi se za direktno pokretanje turbina generatora, te je ovo i najjednostavniji princip korištenja geotermalne energije.

Princip separiranja pare - koristi se vruća voda iz geotermalnih rezervoara pod velikim tlakom i temperaturama iznad 180˚C. Pumpanjem vode iz tih rezervoara prema elektrani na površini smanjuje se tlak, pa se vruća voda pretvara u paru i pokreće turbinu. Voda koja se nije pretvorila u paru vraća se nazad u rezervoare zbog ponovne upotrebe.

Binarni princip - Voda koja se koristi kod binarnog principa je hladnije nego kod ostalih principa. Kod binarnog principa vruća voda se ne koristi za pokretanje turbine, već se koristi za grijanje druge tekućine, koja ima dosta nižu temperaturu isparavanja, a ta druga tekućina pokreće turbinu generatora.

2.2.5. Biomasa

Kruta biomasa je najčešće upotrebljavana direktno kao sagorljivo gorivo. Njeni oblici i izvori sadrže gorivo dobiveno iz drva, biogeni udio iz komunalnog krutog otpada ili neiskorišteni dio ratarskih kultura.

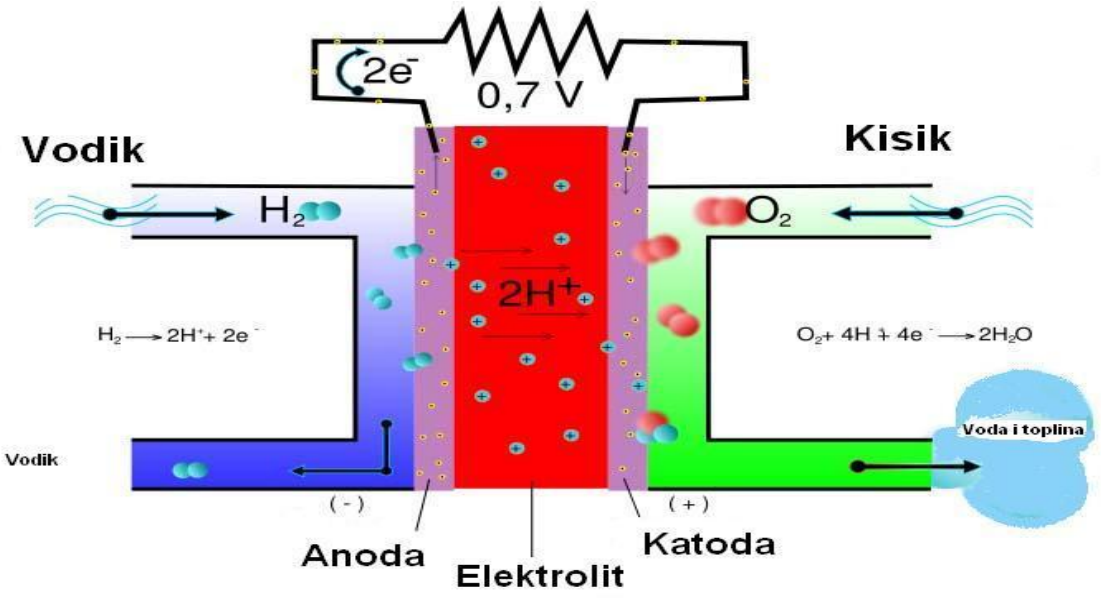
Biomasa se može na različite načine koristiti za proizvodnju električne energije, a najjednostavniji je sagorijevanje biomase u klasičnim elektranama na kruta goriva, koje, radi povećanja korisnosti, najčešće osim električne energije pružaju i toplinsku energiju za grijanje domova te za industriju. Biomasa se, zbog svoje male energetske vrijednosti u odnosu na energiju potrebnu za njen transport, najčešće koristi u blizini izvora biomase.

Biomasa se također može koristiti za proizvodnju bioplina. Bioplin se dobiva anaerobnom razgradnjom organskih tvari, uključujući gnojivo, kanalizacijski mulj, komunalni otpad ili bilo koji drugi biorazgradivi otpad. Dobiveni plin se sastoji od metana i ugljikovog dioksida, te

raznih ostalih plinova, ali u vrlo malim količinama. Bioplin se koristi za pogon plinskih motora koji pokreću električne generatore, te tako proizvode električnu energiju. Energetska vrijednost bioplina je oko 5kWh/Nm^3. Izgaranjem u plinskom motoru dobije se oko 2kWh el. energije. Kod kombinirane upotrebe električne i toplinske energije dobije se puno veći stupanj korisnosti jer se uz 2 kWh električne energije dobije i oko 2.15 kWh toplinske energije.

2.2.6. Gorive ćelije

Gorive ćelije pripadaju elektrokemijskim izvorima el. energije. Elektrokemijski izvori su izvori koji proizvode električnu energiju pomoću elektrokemijskih procesa. Gorive ćelije mogu koristiti razne vrste goriva bogatih vodikom kao što su prirodni plin, benzin, bioplin i propan. Gorive ćelije se sastoje od dvije elektrode koje su odvojene membranom ili elektrolitom. Na anodu se dovodi gorivo koje na njoj oksidira. Elektroni nastali oksidacijom goriva odvode se vodičima preko trošila do katode. Na katodi se neki drugi element (oksidans) reducira zahvatom elektrona proizvedenih na anodi. Proizvodi reakcije, negativni i pozitivni ioni spajaju se u elektrolitu, a nastali produkt odvodi se sa gorive ćelije. Pojednostavljeni prikaz gorive ćelije može se vidjeti na slici 2.3, kod kojega se kao gorivo koristi vodik, a kao oksidans kisik [12].



Slika 2.3. *Goriva ćelija sa vodikom kao gorivom i kisikom kao oksidansom* [15]

Prednosti gorivih ćelija su dosta dobra korisnost (oko 60%), ne sadrže pokretne dijelove, nema sagorijevanja goriva, tako da imaju zanemarivo malu emisiju štetnih plinova (CO_2). Nedostatak gorivih ćelija je što im se starenjem povećava unutarnji otpor, te moraju koristiti elektroničke uređaje za regulaciju izlaznog napona.

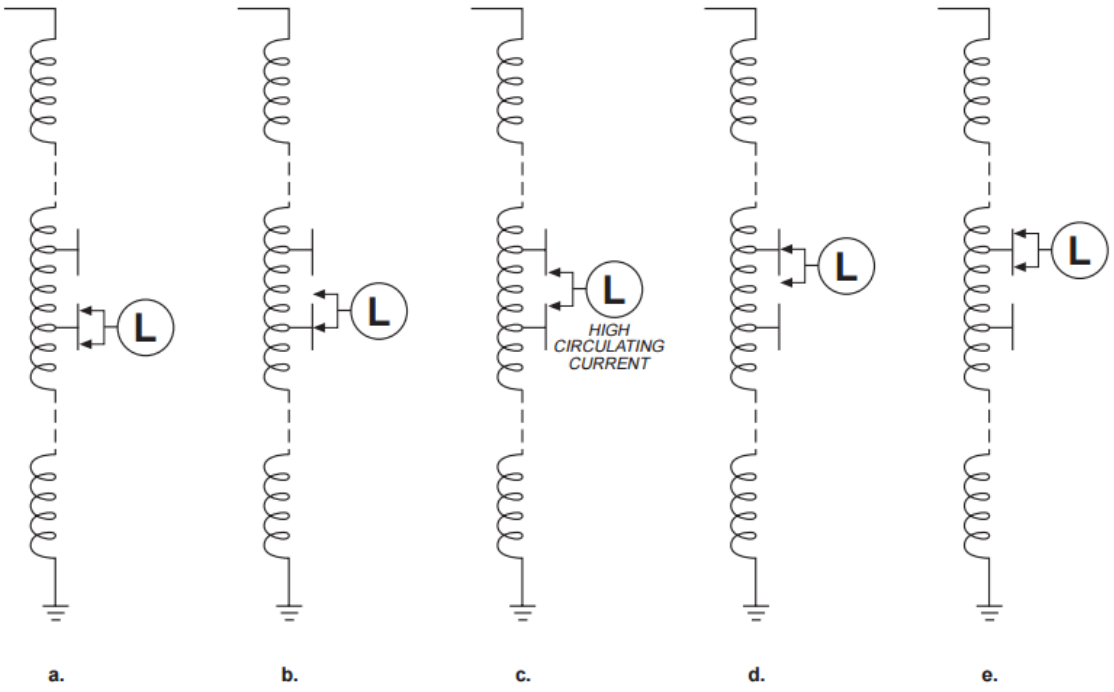
3. TRANSFORMATORI S REGULACIJOM NAPONA

Da bi se regulirao napon na transformatoru potrebno je promijeniti prijenosni omjer transformatora, a to se radi promjenom broja zavoja. Promjena broja zavoja se najčešće radi na primarnoj strani, ali je moguće i na sekundarnoj. Postoje dvije vrste transformatora s regulacijom napona, a to su: transformatori s otcjepima i regulacijski transformatori.

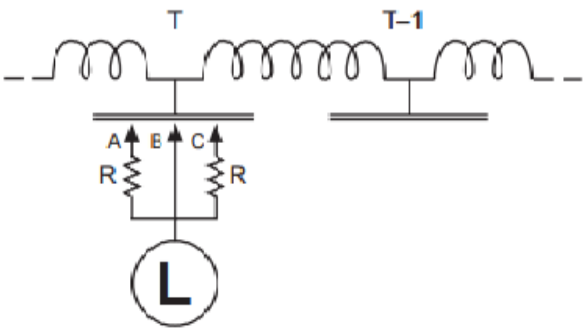
Transformatori s otcjepima prijenosni omjer mogu mijenjati samo u beznaponskom stanju. Otcjepi su uglavnom ±4% ili ±5% no mogu biti i manjeg ili većeg iznosa. Izvedba otcjepa, s obzirom na konstrukciju transformatora je nepovoljna, pa se ona uglavnom izbjegava.

Kod regulacijskih transformatora je moguće omjer mijenjati i za vrijeme pogona. Oni se izvode sa puno većim brojem otcjepa (21, 25...), pa su time koraci regulacije manji(1,5-2%). Regulacijska se preklopka može staviti na VN i NN stranu, no većinom se stavlja na VN stranu, jer su tamo manje struje, pa su sklopke jeftinije i lakše.

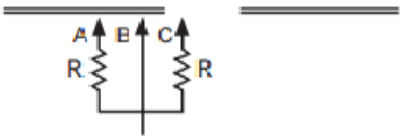
Na slici 3.1. mogu se vidjeti koraci regulacijske preklopke kod promjene omjera transformatora. Na njoj se može primijetiti da u trećem koraku dolazi do kratkog spoja dijela zavoja. Taj problem se rješava dodavanjem radnog ili reaktivnog otpora između dvaju otcjepa.



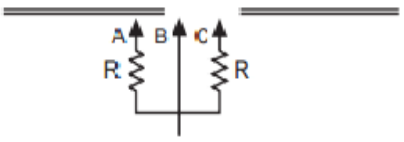
Slika 3.1. *Koraci regulacijske preklopke kod promjene omjera transformatora* [7]



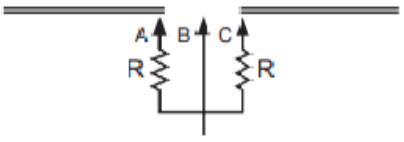
Preklopka na položaju T, sva struja prolazi kroz B.



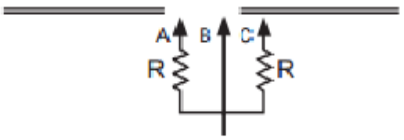
C se odvaja, pošto kroz C ne prolazi struja, nema iskrenja. Sva struja ide kroz B



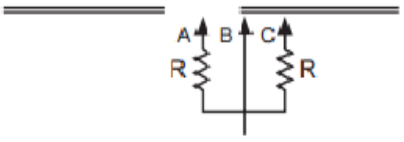
B se odvaja, dolazi do iskrenja kako struja sa B prelazi na A. Napon sekundara se smanjuje za I_{opt}*R.



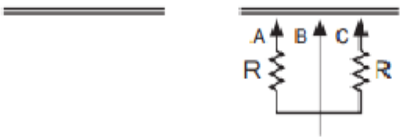
C se spaja na položaj T-1, dolazi do iskrenja. Struja opterećenja dijeli se između A i C.



A se odvaja uz iskrenje. Sva struja prolazi kroz C.



B se spaja, dolazi do iskrenja jer sva struja prelazi na B



A se spaja i promjena položaja preklopke je završena.

Slika 3.2. *Koraci regulacijske preklopke s otporom* [7]

Na slici 3.2. prikazani su koraci kod promjene položaja regulacijske preklopke s ugrađenim otporom. Kod ove preklopke najvažnije je pravilno dimenzionirati otpor, jer ako bi se npr. radilo o transformatoru s korakom od 50V, te ako bi se odabrao otpor od 1 ohm, u otporu bi se moralo potrošiti 2500 W. Otpor također ne smije biti prevelik, jer bi tada na njemu bio prevelik pad napona. Iz ovoga se može zaključiti da se promjena položaja odvija u vrlo kratkom vremenu, koje obično iznosi nekoliko desetaka milisekunde.

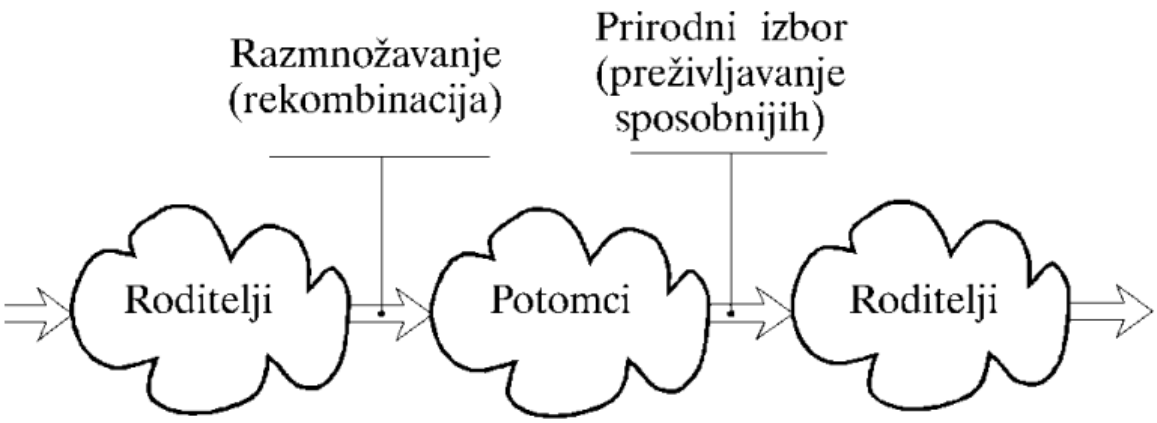
Slično ovome, umjesto otpora, preklopka može imati induktivni otpor. Prednost ovakve preklopke je što su gubici koji nastaju prolaskom struje kroz zavojnicu zanemarivi. Također, preklopka se može nalaziti između dva položaja neograničeno vrijeme, bez ikakvih gubitaka, a napona je tada sredina između 2 položaja preklopke. Zbog ovoga je kod ovakvih preklopki potrebno duplo manje glavnih zavojnica, jer se koriste i položaji između dviju glavnih zavojnica.

Nedostatak ovih preklopki je što kod promjene položaja dolazi do iskrenja. Ovaj problem se rješava dodavanjem vakuumske preklopke, koja se kombinira s induktivnim otporom, zbog manjih gubitaka i potrebnog manjeg broja zavojnica[6].

4. EVOLUCIJSKI ALGORITMI

4.1. Prirodna evolucija

Sredinom 19. stoljeća engleski znanstvenik Charles Darwin postavio je temelje moderne teorije o evoluciji živih bića. Darwinova teorija evolucije kaže kako se mladunčad svake vrste natječe za opstanak. Oni mladi koji prežive kako bi proizveli slijedeću generaciju prisvajaju pogodne prirodne varijacije kroz proces prirodne selekcije i te varijacije prenose na svoje potomke. Stoga se svaka nova generacija "poboljšava" u odnosu na prethodnu generaciju. Umjesto poboljšavanje iz generaciju u generaciju, točnije bi bilo reći da se nove generacije bolje prilagođavaju svom okolišu. Darwin je zamijetio kako živa bića stvaraju više potomaka od cijele njihove populacije. Iz ovoga bi se dalo zaključiti kako bi broj živih bića koje čine populaciju eksponencijalno rastao, ali unatoč tome, broj jedinki populacije je uglavnom konstantan. Razlog tome je prirodna selekcija. Ona kaže kako će unutar populacije preživjeti i stvarati potomke jedinke koje su bolje prilagođene okolini i uvjetima u kojima žive [10]. Ovo je shematski prikazano na slici 4.1.



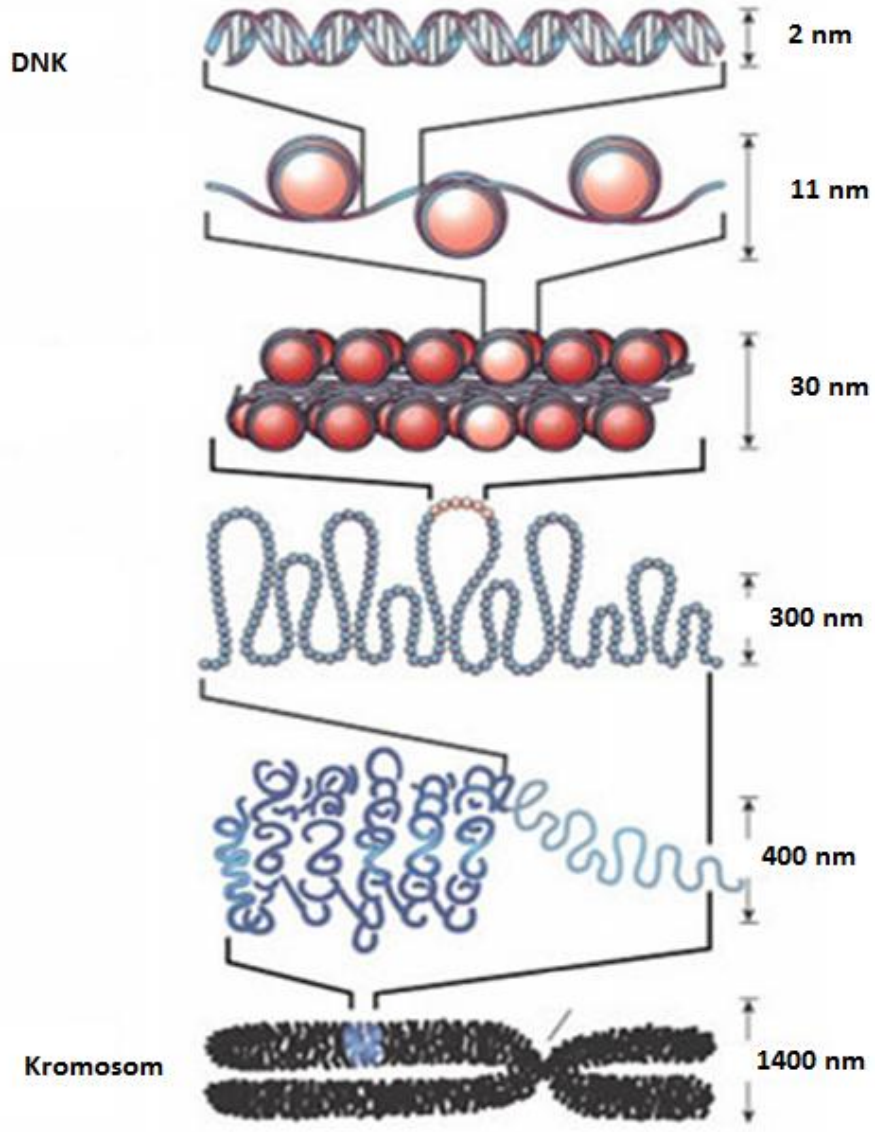
Slika 4.1. *Shematski prikaz evolucije*

4.1.1. Kromosomi i molekula DNK

Kromosomi su lančaste tvorevine koje se nalaze u jezgri svake stanice. Pretpostavlja se da su sva svojstva neke jedinke zapisana u kromosomima. U svaki djelić kromosoma zapisan je skup informacija koje karakteriziraju jedno svojstvo (npr. boja kože, boja očiju, broj zubi i sl.). Taj djelić kromosoma naziva se gen. Kromosom možemo zamisliti kao poredani niz gena. Tada je gen određen svojim položajem u tom nizu i svojim kemijskim sastavom. Kromosomi uvijek dolaze u parovima, gdje je jedan par od oca, a drugi od majke. Za svako svojstvo dakle postoje dva gena. U ovakvom paru geni mogu biti ravnopravni ili neravnopravni. Kod ravnopravnih

parova rezultantno svojstvo se nalazi negdje između svojstva oca i majke, dok kod neravnopravnih prevladava svojstvo dominantnog gena, te taj gen određuje rezultantno svojstvo.

Prema [7], sva svojstva jedinke nisu zapisana u samo jednom paru kromosoma, već u nekoliko desetaka kromosoma (npr. čovjek ima 23 para, vinska mušica 4 para, gorila 24 ...)



Slika 4.2. *Sastav kromosoma* [17]

Na slici 4.2. vidi se izgled i sastav kromosoma. Sastavni dio kromosoma je molekula DNK (Deoksiribonukleinska kiselina). DNK ima oblik dvostruke spiralne zavojnice (Sl. 4.3.) i sastoji se od pentoze, deoksiriboze, fosfatne skupine, i dušičnih baza koje mogu biti adenin (A), timin

(T), gvanin (G) i citozin (C) [11]. Kemijsku strukturu DNK otkrili su Watson i Crick 1953. godine.



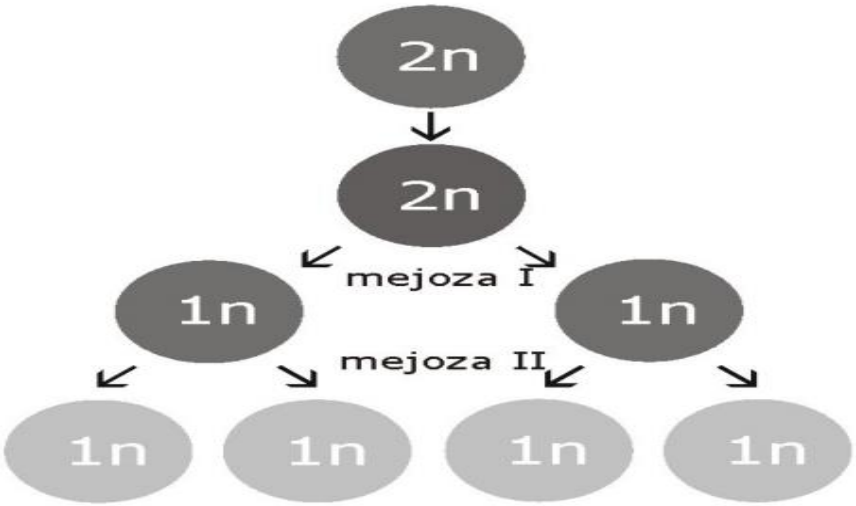
Slika 4.3. *Molekula DNK* [18]

Između 2 lanca DNK svaka baza jednoga može biti spojena s komplementarnom bazom drugoga i to tako da se adenin s dvije vodikove veze spaja s timinom, a gvanin, sa 3 vodikove veze, spaja se s citozinom. Pokazalo se kako su upravo te dušične baze jedinice informacije, kao što je na računalu najmanja jedinica informacije jedan bit (0 ili 1)

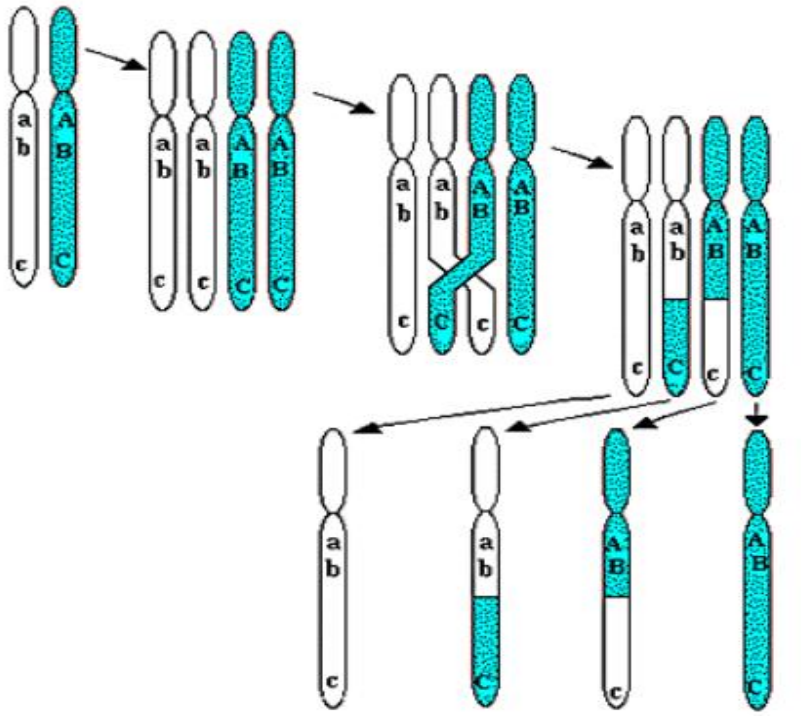
4.1.2. Križanje i mutacija u prirodi

Križanje u prirodi vrlo je složen proces kojim se geni, odnosno svojstva roditelja prenose na djecu. Glavnu ulogu kod križanja imaju kromosomi. Broj kromosoma jednog roditelja je 2n i kaže se da je to diploidan broj kromosoma. Kako bi broj kromosoma djece bio jednak broju kromosoma roditelja, mora doći do mejoze (Sl. 4.4.). Mejoza je stanična dioba u kojoj se broj kromosoma smanjuje na polovinu. Krajnji rezultat mejoze su 4 stanice s polovičnim brojem kromosoma. Za vrijeme mejoze, kromosomi se razmotaju u duge niti te su spojeni u samo jednoj točki i između njih dolazi do izmjene pojedinih segmenata kromosomskih niti kao što pokazuje slika 4.5. . Ovaj proces naziva se križanje ili rekombinacija.

Mutacija nastaje slučajnom promjenom u strukturi DNK. Vjerojatnost mutacije različitih gena se uglavnom kreće od 10^{-4} do 10^{-5}. Mutacijom se svojstva jedinke mogu poboljšati, pa se ta svojstva prijenose na slijedeće generacije. Suprotno ovome, mutacije mogu i pogoršati svojstva jedinke, te bi to moglo dovesti do odumiranja jedinke.



Slika 4.4. *Pojednostavljeni prikaz mejoze*



Slika 4.5. *Križanje ili rekombinacija za vrijeme mejoze* [19]

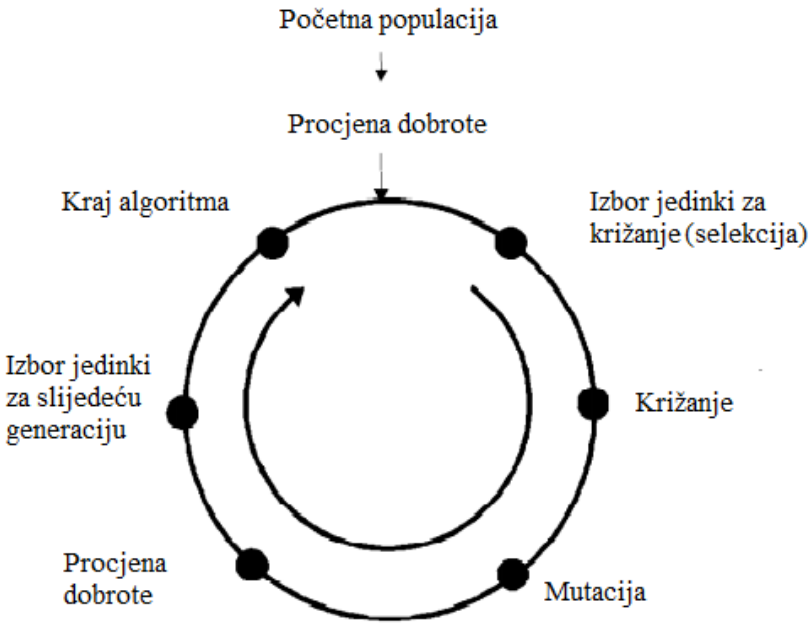
4.2. Vrste evolucijskih algoritama

Evolucijski algoritmi su skup metoda za traženje optimalnog rješenja zadanog problema. Oni u svom radu koriste zakone i procese preuzete iz prirodne evolucije.

Osnovne vrste evolucijskih algoritama su:

- Evolucijska strategija
- Evolucijsko programiranje
- Genetsko programiranje
- Genetski algoritmi
- Klasifikatorski sustavi

Rad evolucijskog algoritma prikazan je na dijagramu sa slike 4.6. . U početnom koraku odabire se populacija, najčešće slučajnim odabirom. Nakon toga se procjenjuje dobrota pojedinih jedinki, te se vrši odabir jedinki koje će sudjelovati u križanju. Nakon odabira, obavlja se križanje, te mutacija nasumično odabranih jedinki. Kada je završen ovaj korak, ponovo se procjenjuje dobrota jedinki, te se odabiru jedinke za slijedeću generaciju. Ovaj se proces odvija sve dok se ne dobije zadovoljavajuće rješenje.



Slika 4.6. *Dijagram rada evolucijskog algoritma* [16]

4.2.1. Jedinka

Jedinke u evolucijskom algoritmu predočene su neovisnim matematičkim objektom (realni broj, vektor, binarni broj ...). Pravilan izbor predodžbe jedinke je nužan jer on bitno utječe na djelotvornost algoritma i mogućnost pronalaženja zadovoljavajućeg rješenja. Četiri su glavne predodžbe jedinki:

- Predodžba jedinke prilagođena zadatku
- Predodžba jedinke prilagođena evolucijskom algoritmu
- Posebna predodžba
- Prilagodljiva predodžba jedinke

Primjer predodžbe jedinke može se pokazati na funkciji:

$$F(x_1, x_2) = 2 + {x_1}^2 - \cos(18 \cdot x_1) - \cos(18 \cdot x_2) \tag{4-1}$$

Ova funkcija ima puno ekstrema, od kojih je globalni ekstrem za vrijednost [0,0] , a jedinke imaju oblik vektora sa 2 elementa, koji su realni brojevi.

Sve jedinke unutar jedne generacije nazivaju se populacijom.

4.2.2. Funkcija dobrote

Funkcija dobrote (eng. Fitness function) ili funkcija procjene kvalitete jedinke uglavnom je jednaka funkciji koju želimo optimirati [8]. Ona je ključna za proces selekcije, jer upravo ona odlučuje koje jedinka zadovoljava zadane uvijete, te koja će jedinka biti odabrana za slijedeću generaciju. Funkcija dobrote mora biti u skladu s ograničenjima. Definiranjem ograničenja, smanjuje se prostor pretraživanja rješenje, te se samim time brže dolazi do rješenja. Dobrota određene jedinke opisuje kvalitetu rješenja dobivenog uporabom te jedinke. Što je dobrota jedinke veća, to je i vjerojatnost njenog preživljavanja veća. Postupak izračunavanja dobrote jedinke zove se procjena dobrote jedinke.

4.2.3. Selekcija

Cilj je selekcije prenošenje dobrih svojstava jedinki na slijedeće generacije. U prirodi preživljavaju one jedinke koje su bolje prilagođene okolišu. Ako mjeru prilagođenosti nazovemo dobrota, tada će veću vjerojatnost preživljavanje i reprodukcije imati jedinke sa većom dobrotom. Selekcija je postupak kojim se određuje koje jedinke će preživjeti, odnosno stvarati

potomstvo. Na ovaj način dobra svojstva prenose se na slijedeću generaciju, dok loša s vremenom izumiru [8].

Postoji više vrsta selekcija, ali se sve zasnivaju na tome, da jedinke sa većom dobrotom imaju veću vjerojatnost preživljavanja. Jedna od njih je turnirska selekcije, koja nasumično odabire dvije jedinke i jedinka s većom dobrotom se odabire za slijedeću generaciju. Ovaj postupak se ponavlja sve dok se ne popuni generacija novih jedinki. Kod ovakve selekcije niti jedna jedinka nema jamstvo da će preživjeti, a na taj način se izbjegava usmjeravanje algoritma ka lokalnom minimumu.

4.2.4. Operatori varijacije

Uloga operatora varijacije je od starih jedinki stvoriti nove. Dva su operatora varijacije kod evolucijskih algoritama, a to su: Križanje i Mutacija

Kod evolucijskih algoritama, križanje je postupak stvaranja novih jedinki na temelju dviju postojećih jedinki. Najvažnija karakteristika križanja je da djeca nasljeđuju svojstva roditelja. Križanjem dvaju roditelja, možemo dobiti jednog ili dva potomka. Postoji više vrsta križanja (s jednom točkom prekida, s dvije točke prekida, uniformno križanje...), ali sva od njih zasnivaju se na tome da djeca dio gena preuzimaju od jednog, a dio gena od drugog roditelja. Ovo se može vidjet i na primjeru sa tablice 4.1. na kojoj je prikazano križanje s jednom točkom prekida.

Tablica 4.1. *Roditelji i djeca kod križanja s jednom točkom prekida*

Roditelj 1	1	1	0	0	0	1	0	1	1	1	0	1	0	1	1	0	0	1
Roditelj 2	1	0	0	1	1	0	1	1	0	1	0	1	1	1	0	0	1	1
Dijete 1	1	1	0	0	0	1	0	1	1	1	0	1	1	1	0	0	1	1
Dijete 2	1	0	0	1	1	0	1	1	0	1	0	1	0	1	1	0	0	1

Druga operator varijacije je mutacija. Za razliku od križanja, mutacija se primjenjuje na jednu jedinku. Odabir jedinki za mutaciju je slučajan, te nije uvjetovan prethodno pronađenim rješenjima. Na mutaciju se može utjecati samo izborom učestalosti mutacije jedinki. Postoji više vrsta mutacija, koje djeluju na jedan ili više gena te ih izmjenjuju. Primjer jednostavne mutacije može se vidjeti u tablici 4.2. gdje je nasumično odabran gen koji će biti izmijenjen.

Tablica 4.2. *Jednostavna mutacija*

Kromosom	1	1	0	0	0	1	0	1	1	**1**	0	1	0	1	1	0	0	1
Mutirani kromosom	1	1	0	0	0	1	0	1	1	**0**	0	1	0	1	1	0	0	1

4.2.5. Algoritmi zasnovani na evoluciji

Osim evolucijskih algoritama postoje i drugi algoritmi i metode zasnovani na evoluciji, a čiji je cilj približenje globalnom optimumu, a neki od njih su:

- Optimizacija kolonijom mrava (eng. Ant colony optimization)
- Algoritam krijesnica (eng. Firefly algorithm)
- Optimizacija rojem čestica (eng. Particle swarm optimization)
- Metoda skakutanja žaba (eng. Leap frog optimization)
- Metoda pretraživanja ptice kukavice (eng. Cockoo search algorithm)
- Algoritam pčela (eng. Bees algorithm)
- Metoda kolonizacije bakterija (eng. Bacterial growth optimization)

5. OPIS PROBLEMA OPTIMIRANJA UPRAVLJANJA PROIZVODNJOM DISTRIBUIRANIH IZVORA I PREKLOPKOM DISTRIBUCIJSKIH TRANSFORMATORA

5.1. Optimiranje općenito

Kada govorimo o optimiranju prvo moramo definirati cilj, do kojega treba doći optimiranjem. Taj cilj, ovisno o sustavu kojeg proučavamo, ovisi o jednoj ili više varijabli tog sustava. Optimiranje je pronalaženje tih varijabli, koje najviše odgovaraju zadanom cilju. Cilj kod optimiranja je najčešće minimiziranje ili maksimiziranje funkcije cilja, pa se može reći da je optimiranje pronalaženje ekstrema funkcije cilja.

Matematički bi ovo mogli zapisati na 2 načina, ovisno tražimo li minimum, ili maksimum funkcije cilja:

$$f(x_1, x_2, x_3, \dots, x_n) \rightarrow min \tag{5-1}$$

ili

$$f(x_1, x_2, x_3, \dots, x_n) \rightarrow max \tag{5-2}$$

Funkcija $f(x_1, x_2, x_3, \dots, x_n)$ je funkcija cilja i ona ovisi o n varijabli. Svaka od tih varijabli može biti ograničena, a ograničenja mogu biti implicitna i eksplicitna. Implicitna ograničenja varijable se zadaju gornjom i donjom vrijednosti te varijable, dok se eksplicitna ograničenja dijele na ograničenja jednadžbi i nejednadžbi.

$$h_i(x) = 0, i = 1, \dots, m \tag{5-3}$$

$$g_j(x) \leq 0, j = 1, \dots, k \tag{5-4}$$

Kako bi funkciju s ograničenja zamijenili funkcijom bez ograničenja koristit ćemo metodu Lagrange-ovih množitelja. Nakon što se ova metoda primjeni na našu funkciju, ona će izgledati ovako:

$$L(x, \lambda, \mu) = f(x) + \sum_{i=1}^{m} \lambda_i h_i(x) + \sum_{j=1}^{k} \mu_j g_j \tag{5-5}$$

Funkcije općenito mogu imati više ekstrema od kojih je samo jedan globalni i u njemu je vrijednost funkcije najmanja ili najveća, a ostali su lokalni. Cilj optimiranja je dakle pronalazak globalnog ekstrema funkcije cilja, zadovoljavajući sva zadana ograničenja varijabli.

Kod optimiranja cilj mogu biti razne stvari, kao što su troškovi proizvodnje, smanjenje emisije štetnih plinova, korisnost procesa, povećanje dobiti i slično. Ovakvo optimiranje, sa samo jednim ciljem, naziva se jednociljno optimiranje. Međutim, optimiranje može biti i višeciljno, gdje je na primjer cilj smanjiti emisiju plinova i povećanje dobiti. Kod višeciljnog optimiranja redovito se događa da su ciljevi konfliktni, te se približavanjem jednom cilju, udaljavamo od drugog. Ovdje dakle ne postoji jedno rješenje, nego se radi kompromis između više ciljeva [7]. Iz ovoga se može zaključiti da je višeciljno optimiranje dosta složenije od jednociljnog, te će se iz tog razloga u ovom radu odraditi samo jednociljno optimiranje.

5.2. Optimiranje upravljanja proizvodnjom distribuiranih izvora i preklopkom distribucijskog transformatora s ciljem smanjenja gubitaka

Funkcija cilja kod ovog optimiranja je smanjenje djelatnih gubitaka u nekoj konfiguraciji mreže i možemo ju zapisati ovako:

$$f(x_1, x_2, x_3, \dots, x_n) = P_g \rightarrow min \tag{5-6}$$

P_g - Ukupni gubici djelatne snage

Kod ovog optimiranja cilj je dakle pronaći iznose varijabli x_1, x_2, x_3, ... , x_n za koje će iznos djelatnih gubitaka biti najmanji. Za izračunavanje gubitaka u mreži potrebno je znati konfiguraciju mreže, parametre elemenata u toj mreži, te opterećenja u čvorovima. Konfiguracija mreže i parametri elementa mreže su konstantni, a opterećenja u čvorovima se mijenjaju s vremenom. No da bi se pojednostavnilo računanje, i opterećenja u čvorovima se mogu smatrati konstantnima kroz određeno vrijeme. Proračun tokova snaga obavlja se pomoću matematičkih iterativnih metoda, a u ovom radu koristi se Newton-Raphsonova metoda. Varijable ovog sustava koje utječu na funkciju cilja su položaj preklopke distribucijskog transformatora, odnosno njegov prijenosni omjer, te iznos proizvodnje distribuiranih izvora. Lokacija distribuiranih izvora je određena i ne mijenja se za vrijeme optimiranja.

Funkcija cilja se dakle može zapisati kao:

$$f(P_1, P_2, P_3, \dots P_i \ \dots, P_n, k_{tr}) = P_g \rightarrow min \tag{5-7}$$

P_i - Izlazna snaga pojedinog distribuiranog izvora

k_{tr} - Položaj preklopke distributivnog transformatora

Varijable u ovom slučaju imaju samo implicitna ograničenja. Za donju granicu proizvodnje pojedinog distribuiranog izvora uzima se 0, dok gornje granice nema, odnosno gornja je granica beskonačnost. Ograničenja položaja preklopke distributivnog transformatora postavljaju se ovisno o tipu transformatora i to tako da obuhvaća minimalnu i maksimalnu vrijednost izlaznog napona transformatora. Osim ovih ograničenja, zadano je i ograničenje napona, a napon treba biti unutar zadanih granica.

$$P_i > 0, i = 1, \dots, n \tag{5-8}$$

P_i - Snaga proizvodnje i-tog distribuiranog izvora

n - ukupan broj distribuiranih izvora

$$V_{min} < V_j < V_{max} \text{ , j=1, ... ,m} \tag{5-9}$$

V_{min} - Najmanji dopušteni napon u mreži

V_i - Iznos napona u i-tom čvoru

V_{max} - Najveći dopušteni napon u mreži

$$k_{min} < k < k_{max} \tag{5-10}$$

k - Prijenosni omjer transformatora

k_{min} - Najmanji prijenosni omjer transformatora

k_{max}- Najveći prijenosni omjer transformatora

S obzirom da optimiranje radimo s genetskim algoritmom, u kojemu će se granice vezane prijenosni omjer transformatora i proizvodnje distribuiranih izvora zadati unutar algoritma, potrebno je, u funkciju cilja, dodati samo ograničenje vezano za napone čvorova.

Nakon što se formula 5-5 primjeni na funkciju cilja dobit ćemo funkciju bez ograničenja koja glasi:

$$F(x) = P_g + \lambda \cdot h(Vi) \tag{5-11}$$

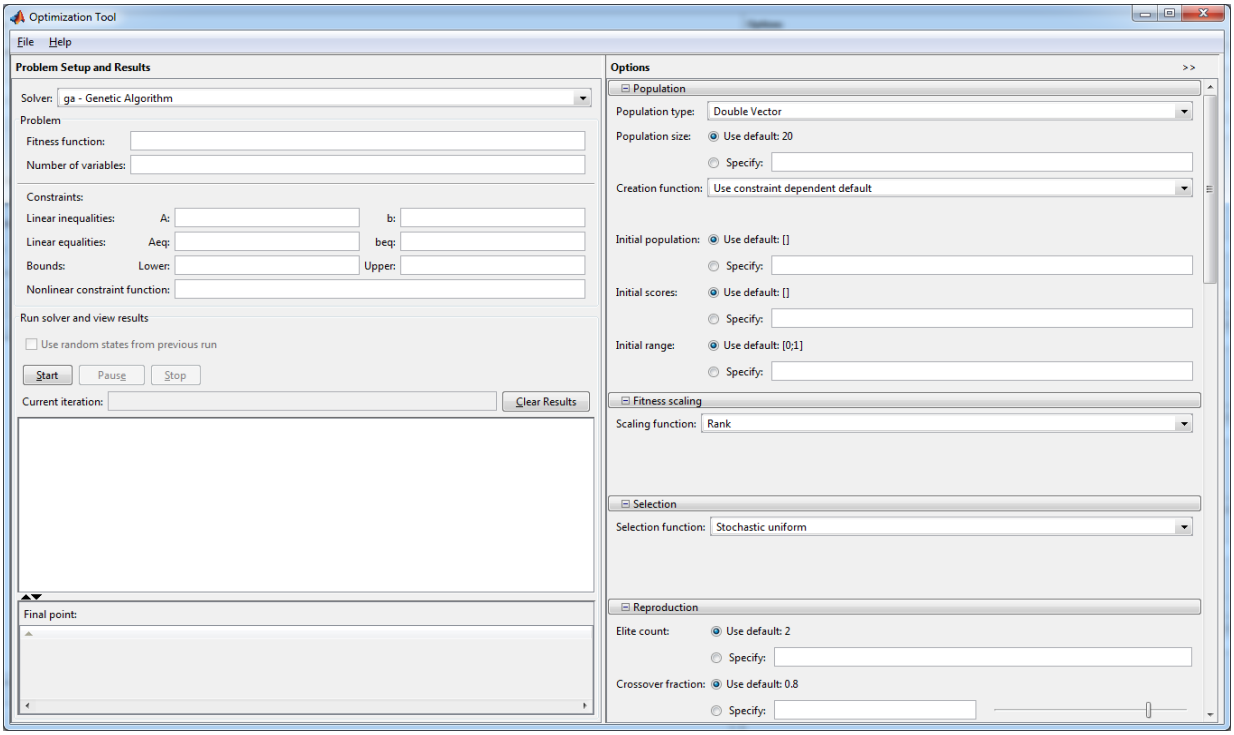
U ovoj formuli koeficijent λ predstavlja koeficijent penalizacije, a funkcija $h(V_i)$ je funkcija penalizacije. Funkcija $h(V_i)$ može se zapisati ovako:

$$\begin{aligned} h(Vi) &= 0 \;\; za\, V_{min} < V_i < V_{max} \\ h(Vi) &= 1 \;\; za\, V_i < V_{min} \;\; ili \;\; V_i > V_{max} \end{aligned} \tag{5-12}$$

Na ovaj način, ako napon bilo kojeg čvora ne zadovolji zadani uvjet, $h(V_i)$ će iznositi 1 i funkcija će se penalizirati, a kod optimiranja će rješenje funkcije cilja biti puno veće, te će vjerojatnost njegovog preživljavanja biti jako mala.

6. OPIS ALATA ZA SIMULACIJU

Za optimizaciju problema u ovom radu korišten je *MATLAB*. MATLAB u sebi ima gotove alate za optimizaciju koji se nalaze u optimizacijskom alatnom okviru. MATLAB sadrži više optimizacijskih metoda, a odabran je genetski algoritam. Optimizacijski alatni okvir unutar MATLAB-a možemo jednostavno pokrenuti upisivanjem komande „*optimtool*" gdje se može odabrati metoda koju želimo koristiti, ili se može direktno otvoriti alat za genetski algoritam upisivanjem komande „ *optimtool "ga"* ". Na slici 6.1. vidi se izgled optimizacijskog alata za genetski algoritam.



Slika 6.6. *Optimizacijski alat za genetski algoritam*

Za pokretanje simulacije nužno je odabrati funkciju cilja (eng. Fitness function), broj varijabli, te ograničenja varijabli, ukoliko postoje. Ukoliko su zadane samo funkcija cilja, broj varijabli te granice, MATLAB će koristiti automatski zadane postavke. Korištenjem automatskih postavki možemo dobiti dobro rješenje, ali to i ne mora biti slučaj. Zbog toga se korištenjem postavki ugađa rad algoritma, kako bi se dobilo što bolje rješenje za određeni problem.

Postavke algoritma koje možemo mijenjati su:

- Populacija
- Skaliranje funkcije cilja

- Selekcija
- Reprodukcija
- Mutacija
- Križanje
- Migracija
- Postavke algoritma
- Hibridna Funkcija
- Kriterij zaustavljanja algoritma
- Funkcije crtanja dijagrama
- Izlazne funkcije
- Prikaz u komandnom prozoru
- Korisnička procjena funkcije

U nastavku će biti detaljnije opisane neke od postavki, koje su se koristile u radu.

6.1. Populacija

Postavkama populacije odabiremo parametre koje genetski algoritam koristi za stvaranje početne populacije. Moguće je odabrati tip populacije, veličinu populacije, funkciju stvaranja, početnu populaciju, početnu dobrotu populacije i početni raspon vrijednosti.

Tip populacije određuje vrstu varijable koju koristi funkcija cilja. Moguće je odabrati:

- **Vektor dvostruke preciznosti** - se koristi kada su jedinke u populaciji takvog tip, ovo je ujedno i zadani tip jedinki
- **Niz bitova** - koristi se kada su jedinke zadane kao niz bitova
- **Proizvoljno** - Kada želimo sami definirati neki drugi tip populacije

Veličina populacije određuje broj jedinki unutar svake populacije. Povećanjem veličine populacije, algoritam detaljnije pretražuje prostor rješenja, te time smanjuje šansu da se algoritam zaustavi na lokalnom ekstremu. No povećanjem broja jedinki, usporava se rad algoritma.

Funkcija stvaranja određuje funkciju koja se koristi za stvaranje početne populacije. Kod odabire funkcije stvaranje imamo 3 opcije:

- **Uniformna** - Stvara nasumičnu početnu populaciju sa uniformnom razdiobom. Ova opcija se koristi kada nisu zadana ograničenja
- **Izvediva populacija** - Stvara nasumičnu populaciju tako da su zadovoljena zadana linearna ograničenja
- **Proizvoljno** - Daje nam izbor da napravimo vlastitu funkciju stvaranja. Kod ovoga se mora paziti da funkcija generira takve jedinke da odgovaraju odabranom tipu populacije.

Početna populacija specificira početnu populaciju, odnosno sami možemo napraviti svoju početnu populaciju. Ako u polje početne populacije unesemo matricu, ona ne smije imati više redova od veličine populacije, i mora imati onoliko stupaca koliko imamo varijabli. Ukoliko se ne unesu podaci za sve jedinke, genetski algoritam poziva funkciju stvaranja, koja tada generira preostale jedinke.

Početna dobrota određuje početnu dobrotu početne populacije.

Početni raspon vrijednosti određuje raspon vektora u početnoj generaciji koju generira funkcija stvaranja.

6.2. Skaliranje funkcije cilja

Skaliranje funkcije cilja pretvara rezultate, koje vraća funkcija cilja, u vrijednosti koje odgovaraju funkciji selekcije. Kod postavki za funkciju skaliranja imamo samo odabir funkcije skaliranja, a ona može biti:

- Rangirajuća
- Proporcionalna
- Top skaliranje
- S linearnim pomakom
- Proizvoljna

Rangirajuća funkcija skaliranja je zadana funkcija. Ona skalira jedinke na temelju njihovog ranga umjesto njihove dobrote. Rang jedinke je njena pozicije u sortiranim vrijednostima neobrađenih dobrota. Dobrota jedinke se nakon toga računa kao $1/\sqrt{r}$ gdje r predstavlja rang jedinke. Dakle, jedinka sa rangom jedan će imati vrijednost dobrote 1, slijedeća $1/\sqrt{2}$ i tako dalje.

Proporcionalna funkcija skaliranja skalira vrijednosti proporcionalno njihovim neobrađenim vrijednostima dobrote.

Top funkcija skaliranja dodaje najboljim jedinkama istu vrijednost. Odabirom ove funkcije otvara se novo polje, količina, koje određuje broj jedinki kojima se dodjeljuje pozitivna vrijednost dobrote. Količina može biti zadana kao cijeli broj između 1 i veličine populacije, ili kao razlomak između 0 i 1. Zadana vrijednost količine iznosi 0.4. Odabrane jedinke dobivaju pozitivnu vrijednost dobrote, a ostale jedinke vrijednost 0.

Skaliranje s linearnim pomakom skalira jedinke na način da najbolja jedinka ima konstantnu vrijednost. Vrijednost najbolje jedinke dobije se množenjem odabrane konstante s prosječnom dobrotom jedinki. Konstanta se odabire u polju maksimalna vjerojatnost preživljavanja koje se pojavljuje prilikom odabira ovog skaliranja.

Proizvoljno skaliranje daje nam mogućnost da sami stvorimo funkciju skaliranja.

6.3. Selekcija

Postavke selekcije određuju na koji način genetski algoritam odabire roditelje za slijedeću generaciju. 88

Postavke koje se mogu odabrati su:

- Stohastička uniformna
- Ostatak
- Uniformna
- Rulet
- Turnirska
- Proizvoljna

Stohastička uniformna funkcija je zadana funkcija. Roditelji se određuju tako da se sve jedinke poredaju na liniji gdje je dužina jedinke proporcionalna njenoj dobroti. Nakon toga algoritam se pomiče u jednakim koracima, te u svakom koraku odabire roditelja na kojem je stao. Veličina koraka određuje se tako da se ukupna dobrota svih jedinki podjeli sa brojem roditelja potrebnih za slijedeću generaciju.

Ostatak je funkcija koja odabire roditelje prema cijelom broju njihove dobrote, i nakon toga koristi rulet selekciju za ostatak dobrote. Primjer za ovo je kada je dobrota jedinke 2,3 , ona će se kao roditelj pojaviti 2 puta. Nakon što se taj postupak napravi za svaku jedinku, ostatak roditelja koji je potreban odabire se stohastički, a vjerojatnost da jedinka bude odabrana za roditelja jednaka je ostatku dobrote, u ovom slučaju je to 0,3.

Uniformna selekcija odabire roditelje na temelju njihove dobrote te broja potrebnih roditelja. Uniformna selekcija nije dobra za pronalaženje dobrog rješenje, već se najčešće koristi za testiranje algoritma.

Rulet selekcija odabire roditelje tako da simulira rulet kotač na kojemu je veličina određenog polja proporcionalna vjerojatnosti odabira te jedinke za roditelja. Algoritam koristi nasumične brojeve kako bi odabrao određeno polje.

Turnirska selekcija odabire roditelje na način da nasumično odabere nekoliko jedinki, te za roditelja odabire jedinku s najvećom dobrotom. Broj jedinki koje ulaze u turnir može se zadati i mora biti 2 ili više. Ukoliko se ne odabere broj jedinki za turnir, koristi se 4 kao zadano.

6.4. Reprodukcija

Postavke reprodukcije određuju kako genetski algoritam stvara jedinke za slijedeću generaciju.

Moguće je postaviti broj elitnih jedinki. Broj elitnih jedinki nam govori koliko će jedinki sigurno preživjeti i prijeći u slijedeću generaciju. Broj jedinki zadaje se kao cijeli broj manji od veličine populacije.

Udio križanja nam govori koliko se jedinki za slijedeću generaciju stvori križanjem. Zadaje se kao razlomak između 0 i 1. Zadana vrijednost udjela križanja je 0.8.

6.5. Mutacija

Postavke mutacije određuju kako će genetski algoritam malim nasumičnim promjenama jedinki stvoriti mutiranu djecu. Svrha mutacije je kako bi se proširilo pregledavanje prostora rješenja, odnosno da bi se spriječilo zaustavljanje algoritma u lokalnom minimumu.

Moguće je odabrati tri vrste mutacije, Gaussova, uniformna, adaptivna izvediva, ali je moguće napraviti i vlastitu funkciju mutacije.

Gaussova funkcija mutacije koristi se za probleme bez ograničenja. Ona svakom elementu jedinke dodaje slučajan broj iz Gaussove distribucije. Standardna devijacije određena je parametrima skaliranja (eng. Scale) i skupljanja (eng. Shrink). Parametar skaliranje određuje standardnu devijaciju početne generacije, a parametar skupljanja određuje koliko se standardna devijacije skuplja iz generacije u generaciju. Ako npr. za parametar skupljanja odaberemo 1, standardna devijacije će se linearno skupljati, dok ne dođe do zadnje generacije gdje će iznositi nula.

Uniformna funkcija mutacije sastoji se od dva koraka. U prvom koraku se odabiru elementi svake jedinke za mutaciju, tako da svaki element ima jednaku vjerojatnost mutiranja, određenu parametrom mjere (eng. rate). U drugom koraku vrijednost elemenata odabranih za mutaciju se zamjenjuje nasumičnim brojem koji se nalazi unutar zadanih granica tog elementa.

Adaptivna izvediva funkcija mutacije koristi se kada postoje ograničenja. Ona nasumično odabire smjerove mutacije koji su adaptivni u odnosu na prethodnu generaciju.

6.6. Križanje

Postavke križanja određuju na koji način se dvije jedinke, odnosno roditelji , kombiniraju kako bi se dobila djeca za slijedeću generaciju.

Raspršeno križanje je zadano ukoliko nisu zadana linearna ograničenja. Kod njega se generira nasumični binarni vektor, te od prvog roditelja uzima gene koje se nalaze na mjestu jedinica tog vektora, a od drugog roditelje gene koje se nalaze na mjestu nula.

Križanje s jednom točkom prekida radi na način da se nasumično odabere broj n koji se nalazi između nule i broja varijabli, te prvih n varijabli odabire od jednog roditelja, a ostatak od drugog.

Križanje s dvije točke prekida je slično križanju sa jednom točkom, samo se ovaj puta odabiru 2 broja, n i m, koje se nalaze između nule i broja varijabli. Nakon toga se od prvog roditelju uzima prvih n varijabli, nakon toga od drugog roditelja, varijable od n do m i na kraju ostatak varijabli ponovo uzima od prvog roditelja.

Posredna funkcija križanja koristi se kada su zadana linearna ograničenja. Kod ovakve funkcije djeca se stvaraju procijenjenim prosjecima roditelja. Ovakvo križanje kontrolira se parametrom "omjer".

Heuristička funkcija križanja stvara potomke koji se nalaze na liniji između dva roditelja i to tako da su bliže roditelju s većom dobrotom. Parametrom omjera određujemo koliko je potomak udaljen od roditelja s većom dobrotom.

Aritmetička funkcija križanja stvara potomke koji su aritmetička sredina oba roditelja.

6.7. Migracija

Migracija određuje kako jedinke migriraju između pod-populacija. Do migracije dolazi ako je za veličinu populacije odabran vektor dužine veće od 1. Kada se dogodi migriranje, najbolje jedinke

iz jedne pod-populacije zamjene najgore jedinke iz druge pod-populacije. Jedinke koje su migrirale se samo kopiraju, odnosno one i dalje postoje u svojoj pod-populaciji.

Parametri koji se mogu postaviti kod migracije su: Smjer, interval i frakcija.

Kako se pod-populacije odnose na formu paralelnog procesiranja, genetski algoritam trenutno ne podržava ovu postavku.

6.8. Hibridna funkcija

Hibridna funkcija je još jedna funkcija optimiranja koja se pokreće nakon što završi genetski algoritam. Ovdje možemo odabrati slijedeće funkcije optimiranja:

- "fminsearch" - koristi MATLAB-ovu funkciju "*fminsearch*" za rješavanje problema bez ograničenja
- "patternsearch" - koristi se za rješavanje problema sa i bez ograničenja
- "fminunc" - koristi funkciju "*fminunc*" iz optimizacijskog alatnog okvira. Koristi se za rješavanje problema bez ograničenja
- "fmincon" - koristi funkciju "fmincon" iz optimizacijskog alatnog okvira. Koristi si za rješavanje problema s ograničenjima

6.9. Kriterij zaustavljanja algoritma

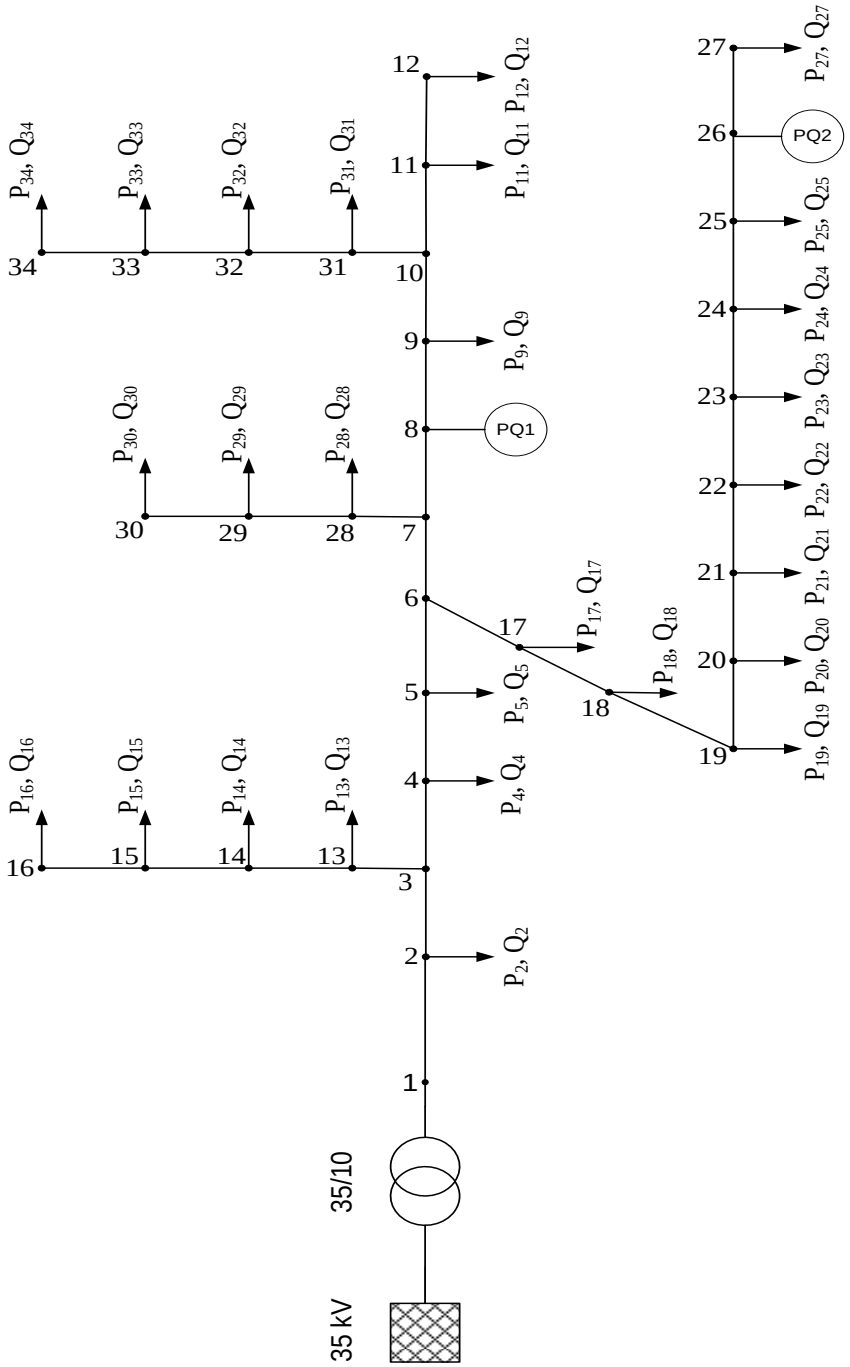
Kriterij zaustavljanja algoritma određuje razlog zaustavljanja algoritma, on može biti:

- Generacije - određuje broj generacija nakon kojeg se algoritam zaustavlja
- Vremensko ograničenje - određuje trajanje rada algoritma
- Ograničenje dobrote - algoritam se zaustavlja ako je vrijednost najbolje dobrote manja ili jednaka postavljenoj vrijednosti
- Otezajuće generacije - algoritam se zaustavlja ako je srednja relativna promjena iznosa dobrote najbolje jedinke, tokom otezajućih generacija manja ili jednaka toleranciji funkcije.
- otezanje vremenskog limita - algoritam se zaustavlja ako tokom zadanog vremena nema poboljšanja dobrote najbolje jedinke
- Tolerancija ograničenja - Tolerancija ograničenja ne koristi se kao kriterij zaustavljanja, već se koristi za određivanje izvodljivosti algoritma s obzirom na nelinearna ograničenja.

Postoje i dodatne postavke algoritma, kao što su prikaz grafova i rješenja algoritma, no s obzirom da ne utječu na rad algoritma, njih ćemo izostaviti.

7. OBAVLJANJE SIMULACIJE

U ovom djelu rada odabrana je 10 kilovoltna mreža sa slike 7.1. te je na njoj ispitan rad genetskog algoritma.



Slika 7.1. *Primjer mreže sa 34 čvora*

Tablica 7.1. *Podaci o impedancijama dionica i nazivnim opterećenjima čvorova za mrežu sa slike 7.1.*

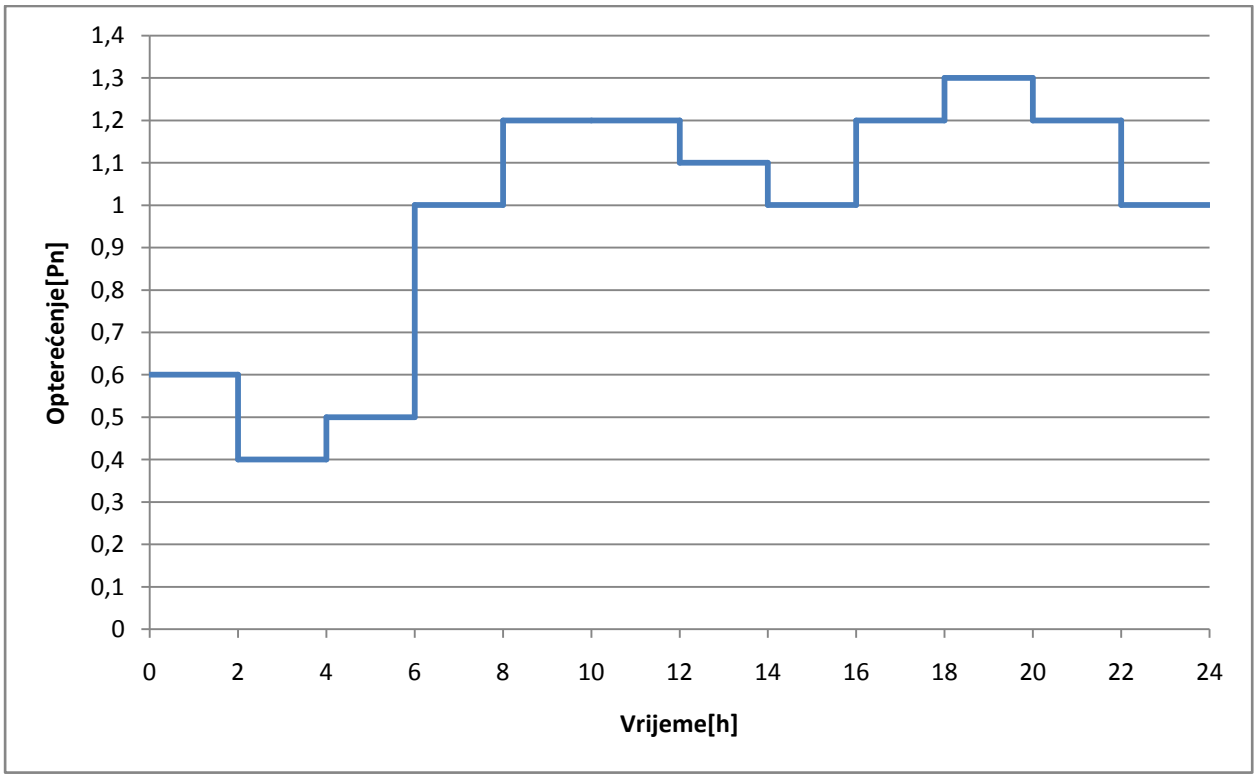
Dionica	Serijska impedancija dionice		Čvor	Snaga opterećenja	
	R [Ω]	X_L [Ω]		P [kW]	Q [kvar]
1 – 2	0,117	0,048	2	230	142,5
2 – 3	0,107	0,044	3	0	0
3 – 4	0,164	0,046	4	230	142,5
4 – 5	0,15	0,042	5	230	142,5
5 – 6	0,15	0,042	6	0	0
6 – 7	0,314	0,054	7	0	0
7 – 8	0,21	0,036	8		
8 – 9	0,314	0,054	9	230	142,5
9 – 10	0,21	0,036	10	0	0
10 – 11	0,131	0,023	11	230	142,5
11 – 12	0,105	0,018	12	137	84
3 – 13	0,157	0,027	13	72	45
13 – 14	0,21	0,036	14	72	45
14 – 15	0,105	0,018	15	72	45
15 – 16	0,052	0,009	16	13,5	7,5
6 – 17	0,179	0,05	17	230	142,5
17 – 18	0,164	0,046	18	230	142,5
18 – 19	0,213	0,047	19	230	142,5
19 – 20	0,194	0,043	20	230	142,5
20 – 21	0,194	0,043	21	230	142,5
21 – 22	0,262	0,045	22	230	142,5
22 – 23	0,262	0,045	23	230	142,5
23 – 24	0,314	0,054	24	230	142,5
24 – 25	0,21	0,036	25	230	142,5
25 – 26	0,131	0,023	26		
26 – 27	0,105	0,018	27	137	85
7 – 28	0,157	0,027	28	75	48
28 – 29	0,157	0,027	29	75	48
29 – 30	0,157	0,027	30	75	48
10 – 31	0,157	0,027	31	57	34,5
31 – 32	0,21	0,036	32	57	34,5
32 – 33	0,157	0,027	33	57	34,5
33 – 34	0,105	0,018	34	57	34,5

U tablici 7.1. nalaze se podaci mreže koja je korištena za ispitivanje genetskog algoritma. Podaci transformatora iz mreže su: S_n=8MVA, u_k=8%, P_k=38 kV. Korišteni transformator im mogućnost promjene napona pod opterećenjem, a preklopka transformatora ima 6 koraka i može mijenjati napon u rasponu od ±5%. Točni koraci promjene napona transformatora mogu se vidjeti u tablici 7.3.

Tablica 7.3. *Položaji preklopke distribucijsko transformatora*

Položaj preklopke transformatora	-3	-2	-1	0	1	2	3
Napon sekundara transformatora	9,5 kV	9,67 kV	9,84 kV	10 kV	10,17 kV	10, 34 kV	10,5 kV

Opterećenje mreže mijenja se tokom dana. U stvarnosti se opterećenje kontinuirano mijenja, ali zbog jednostavnosti, u našem primjeru, opterećenje će se mijenjati u koracima, a dijagram opterećenja vidljiv je na slici 7.2.



Slika 7.2. *Dnevni dijagram opterećenja mreže*

Tehnologija proizvodnje energije koja je povoljna za ovakav način optimiranja je korištenje biomase. Razlog tomu je što ostale vrste obnovljivih izvora ne mogu kontrolirati proizvodnju ovisno o potrebama potrošača, dok je kod biomase to moguće.

Prilikom svakog koraka algoritma potrebno je odraditi proračun tokova snaga. Za proračun tokova snaga korišten je PSAT (eng. Power System Analysis Toolbox) koji je dostupan kao dodatak MATLAB-u. Za svaki proračun, korištena je Newton-Raphsonova metoda.

7.2. Parametri genetskog algoritma

Prije nego što se ispita rad algoritma za mrežu, potrebno je odabrati optimalne parametre genetskog algoritma, odnosni odabrati takve parametre s kojima će se dobiti najbolji mogući rezultat.

7.2.1. Veličina populacije

U ovom dijelu ispituje se utjecaj veličine populacije na točnost rješenja. Ispitivanje će se raditi od 10 do 100 jedinki. Ostale postavke algoritma su:

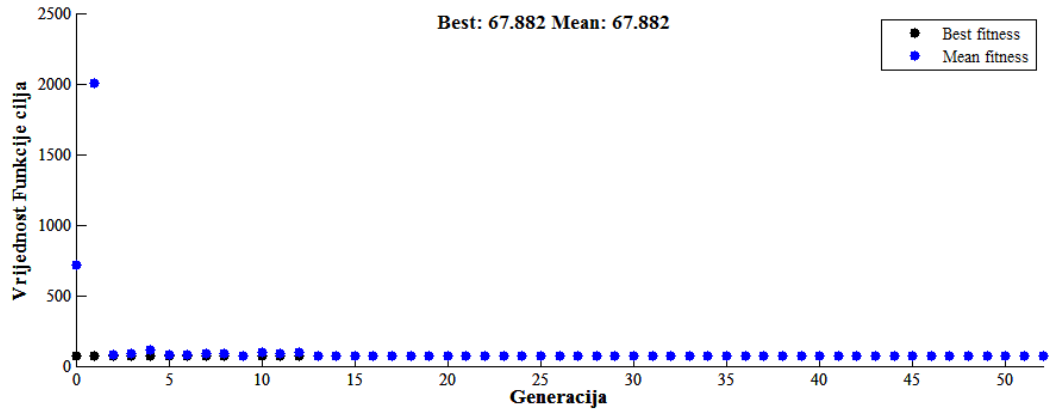
- Skaliranje: Top skaliranje
- Funkcija selekcije: Turnirska
- Broj elitnih jedinki: Dvije
- Udio križanja: 0.8
- Križanje: Raspršeno

Tablica 7.4. *Snaga gubitaka za različite iznose veličine populacije*

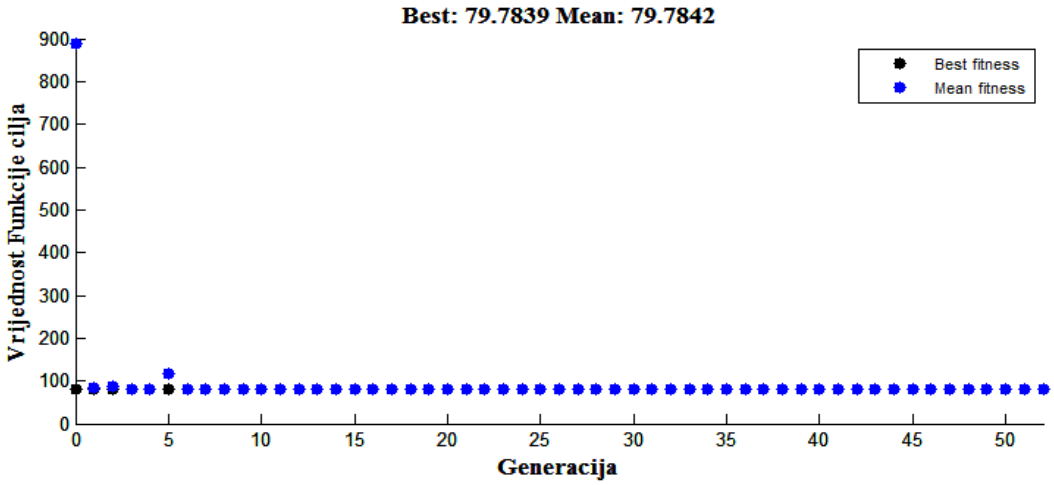
Veličina populacije	Snaga gubitaka [kW]				Srednja vrijednost
10	79,78394	67,88227	70,1692	70,76497	72,150095
20	67,8823	67,88482	67,88344	67,88197	67,883134
30	67,882	67,88198	67,882	70,1669	68,453216
50	67,88197	67,88206	67,88234	67,88197	67,882085
100	67,88197	67,88197	67,88197	67,88197	67,881969

Za svaku veličinu populacije, proračun se izvodio 4 puta. Iz tablice je vidljivo da se povećanjem populacije povećava i točnost rješenja, osim za slučaj sa 30 jedinki. Daljnjim povećavanjem veličine populacije povećava se vrijeme izvođenja algoritma, a rješenja se razlikuju vrlo malo.

Najbolji rezultat dobiven je s veličinom populacije od 100 jedinki, no kako je razlika svega 0.117 W, a vrijeme izvođenja je 2 puta veće, za daljnje izvođenje može se odabrati veličina populacije od 50 jedinki. Sa slika 7.3 i 7.4. vidi se da neovisno o veličini populacije, rješenje vrlo brzo konvergiraju ka konačnom, a razlika je samo u točnosti rješenja.



Slika 7.3. *Prikaz vrijednosti najbolje jedinke i srednje vrijednosti svih jedinki po generacijama za veličinu populacije od 50 jedinki*



Slika 7.4. *Prikaz vrijednosti najbolje jedinke i srednje vrijednosti svih jedinki po generacijama za veličinu populacije od 10 jedinki*

7.2.2. Funkcija skaliranja

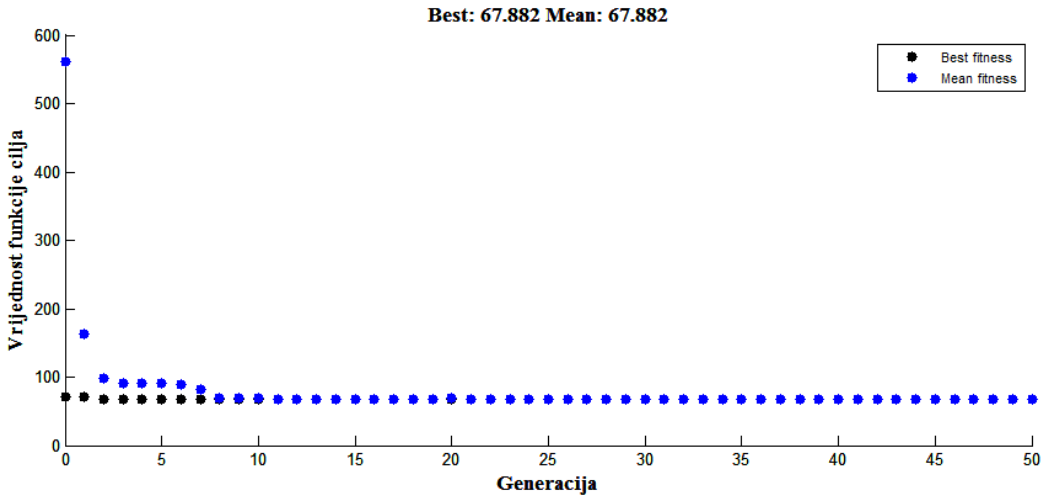
U prethodnom potpoglavlju korištena je top funkcija skaliranja. U ovom potpoglavlju obavit će se proračun i sa ostalim funkcijama skaliranja, kako bi se odredilo koja od njih daje najbolje rezultate za dani problem. Postavke algoritma kod proračuna su slijedeće:

- Veličina populacije: 50
- Funkcija selekcije: Turnirska
- Broj elitnih jedinki: 2
- Udio križanja: 0.8
- Križanje: Raspršeno

Tablica 7.5. *Snaga gubitaka za različite funkcije skaliranja*

Funkcija skaliranja	Snaga gubitaka (kW)				Srednja vrijednost
Rangirajuća	67,88197	67,88197	67,88197	67,88197	67,881971
Proporcionalna	67,88197	67,88197	67,88199	67,88197	67,881975
Top skaliranje	67,88197	67,88206	67,88234	67,88197	67,882085
S linearnim pomakom	67,88197	67,88197	67,88197	67,88197	67,881970

Iz tablice se može vidjeti da za ovaj primjer mreže, funkcija skaliranja nema velik utjecaj. No kako je za funkciju skaliranja s linearnim pomakom dobiven najbolji rezultat, ona će se koristiti u daljnjim proračunima.



Slika 7.5. *Prikaz najbolje jedinke i srednje vrijednosti svih jedinki po generacijama za skaliranje s linearnim pomakom*

Sa slike 7.5. vidljivo je kako rješenja vrlo brzo konvergiraju, te su već kod 10. generacije vrlo blizu konačnom rješenju.

7.2.3. Funkcija selekcije

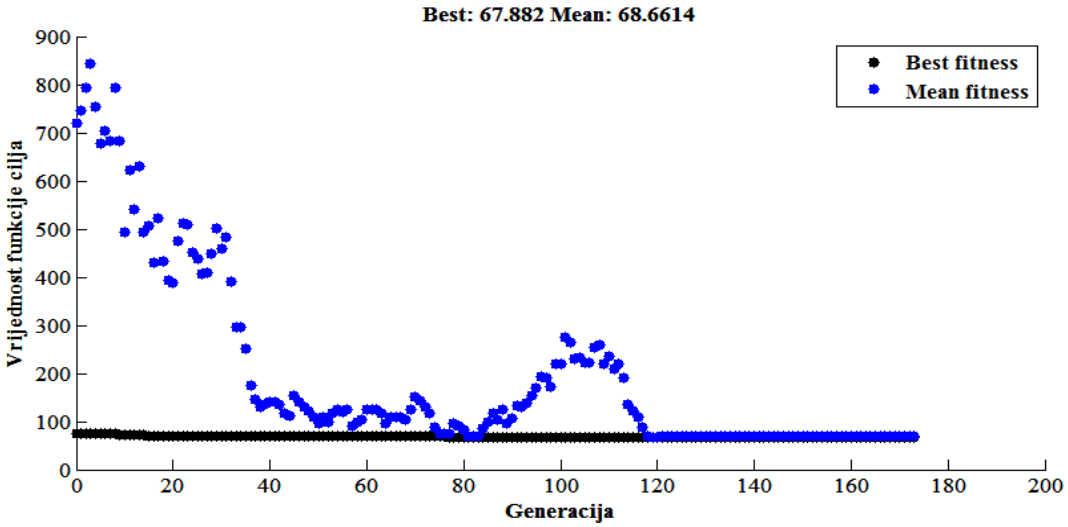
U prethodnim potpoglavljima koristila se turnirska funkcija selekcije, a u ovome će se ispitati kako na rješenje algoritma utječe funkcija selekcije. Ostale postavke algoritma su:

- Veličina populacije: 50
- Skaliranje: S linearnim pomakom
- Broj elitnih jedinki: Dvije
- Udio križanja: 0.8
- Križanje: Raspršeno

Tablica 7.6. *Snaga gubitaka za različite funkcije selekcije*

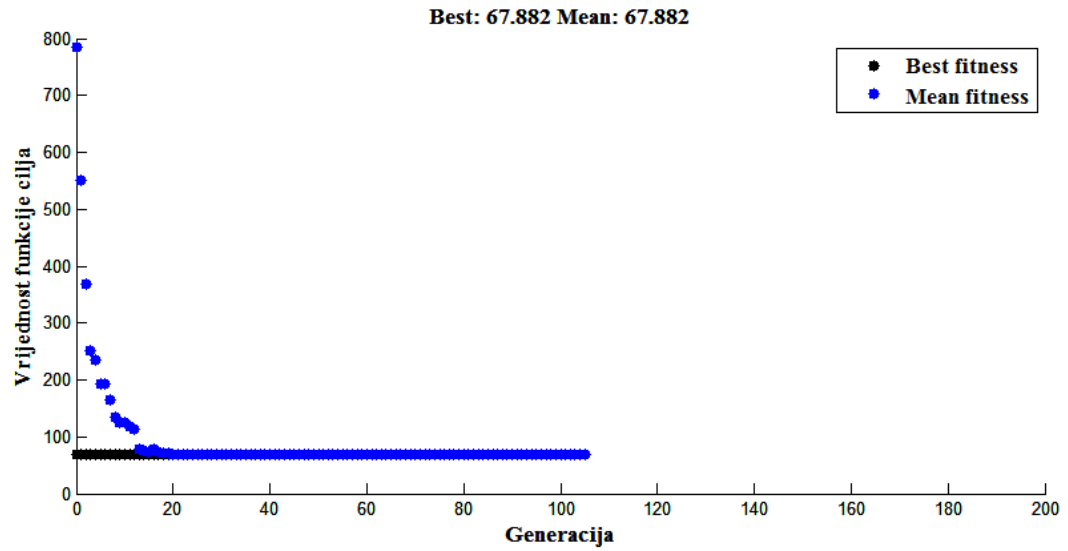
Selekcija	Snaga gubitaka (kW)				Srednja vrijednost
Stohastička uniformna	67,88691	67,91684	67,88517	67,89962	67,897137
S ostatkom	70,16915	67,88810	67,88198	68,02137	68,490152
Uniformna	68,85682	70,40964	68,70742	67,93944	68,978330
Rulet	67,88439	67,88435	67,90050	71,35074	68,754996
Turnirska	67,88197	67,88197	67,88197	67,88197	67,881970

Iz tablice 7.6. vidimo kako je turnirska selekcija, koja je i do sada korištena, dala najbolje rezultate, te će se ona koristiti u daljnjim ispitivanjima. Također se vidi kako odabir funkcije selekcije ima veći utjecaj, u odnosu na prošlo potpoglavlje, gdje su se rješenja vrlo malo razlikovala.



Slika 7.6. *Prikaz najbolje jedinke i srednje vrijednosti svih jedinki po generacijama za uniformnu selekciju*

Na slikama 7.6. i 7.7. mogu se vidjeti srednja vrijednost i vrijednost najbolje jedinke za uniformnu i stohastičku uniformnu funkciju selekcije. Iz prošlog poglavlja vidi se kako za turnirsku funkciju selekcije srednja vrijednost konvergira već oko 10. generacije. Kod stohastičke uniformne funkcije rješenja konvergiraju oko 20. generacije, a kod uniformne tek kod 120. generacije.



Slika 7.7. *Prikaz najbolje jedinke i srednje vrijednosti svih jedinki po generacijama za stohastičku uniformnu selekciju*

7.2.4. Križanje

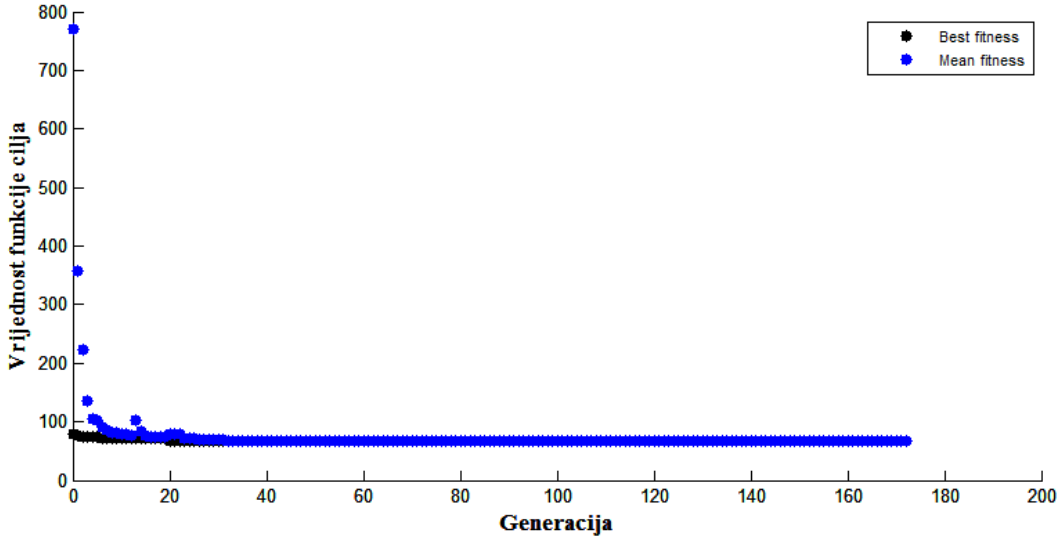
U ovom potpoglavlju ispituje se utjecaj odabira funkcije križanja na konačno rješenje algoritma. Kao i u prethodnim potpoglavljima, za svaku od funkcija proračun će se odraditi četiri puta. Postavke algoritma su:

- Veličina populacije: 50
- Skaliranje: S linearnim pomakom
- Funkcija selekcije: Turnirska
- Broj elitnih jedinki: 2
- Udio križanja: 0.8.

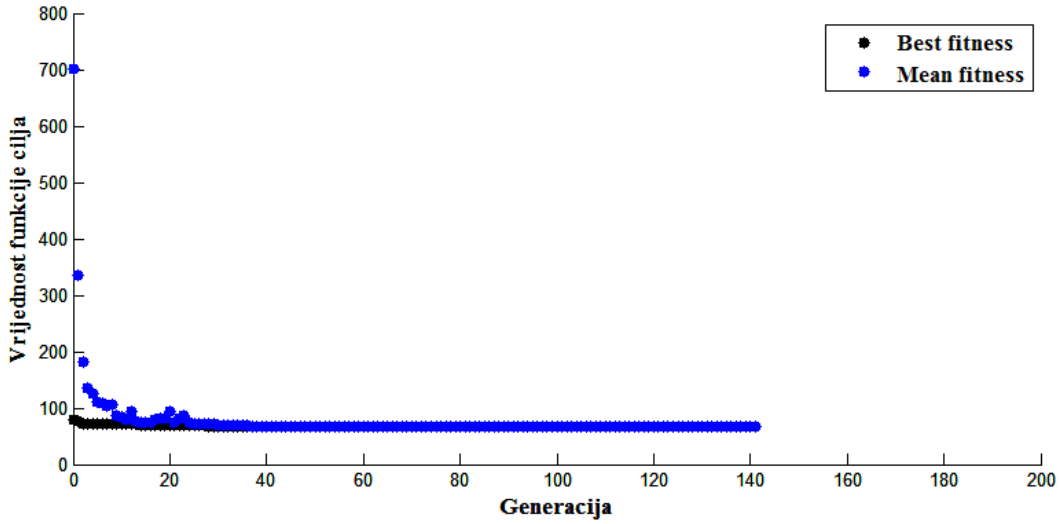
Tablica 7.7. *Snaga gubitaka za različite funkcije križanja*

Križanje	Snaga gubitaka (kW)				Srednja vrijednost
Raspršeno	67,88197	67,88197	67,88197	67,88197	67,881968
S jednom točkom prekida	67,88204	67,88198	67,88197	67,88200	67,88200
S dvije točke prekida	67,88204	68,06683	67,88205	67,88258	67,928378
Posredno	67,88197	67,88358	67,88197	67,88295	67,882615
Heurističko	67,88216	67,88198	67,88535	67,88486	67,883589
Aritmetičko	67,88197	67,88528	67,9081	70,16664	68,460497

Iz tablice 7.7. može se vidjeti da odabir križanja utječe na rezultate, no razlika u rješenjima nije znatna. Najlošiji rezultat dobio se je korištenjem aritmetičke funkcije križanja, a najbolje rezultate dalo je raspršeno križanje.



Slika 7.8. *Prikaz najbolje jedinke i srednje vrijednosti svih jedinki po generacijama za posredno križanje*



Slika 7.9. *Prikaz najbolje jedinke i srednje vrijednosti svih jedinki po generacijama za aritmetičko križanje*

Na slikama 7.8. i 7.9. prikazane su srednje vrijednosti i vrijednosti najboljih jedinki po generacijama za posredno i aritmetičko križanje. U oba slučaja srednja vrijednost jedinki konvergira oko 30. generacije.

7.2.5. Broj elitnih jedinki

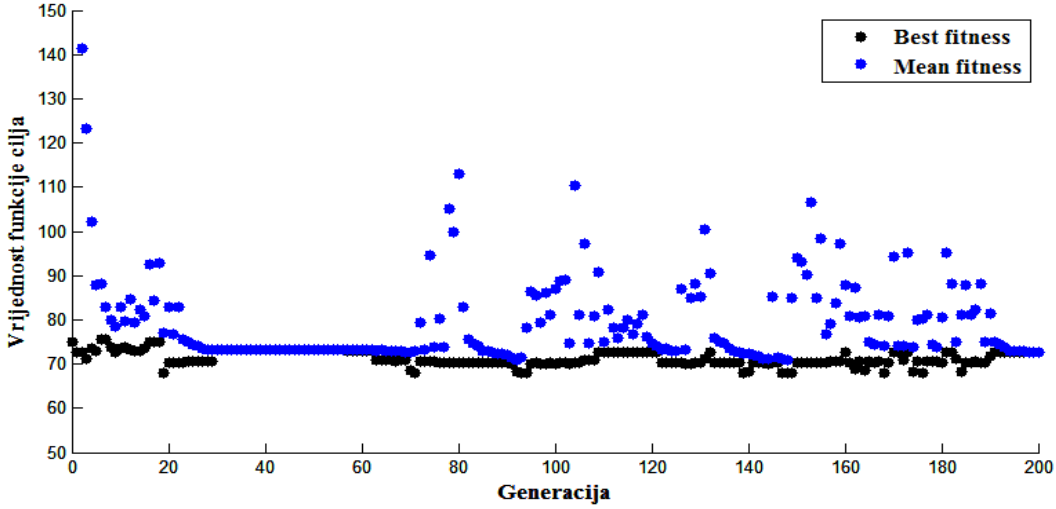
U ovom potpoglavlju ispitan je utjecaj broj elitnih jedinki na konačno rješenje algoritma. Proračun se obavio za slučajeve sa 0, 1, 2, 3 i 4 elitne jedinke, a rezultati su vidljivi u tablici 7.8. Postavke algoritma su:

- Veličina populacije: 50
- Skaliranje: S linearnim pomakom
- Funkcija selekcije: Turnirska
- Udio križanja: 0,8
- Križanje: Raspršeno

Tablica 7.8. *Snaga gubitaka za različite vrijednosti broja elitnih jedinki*

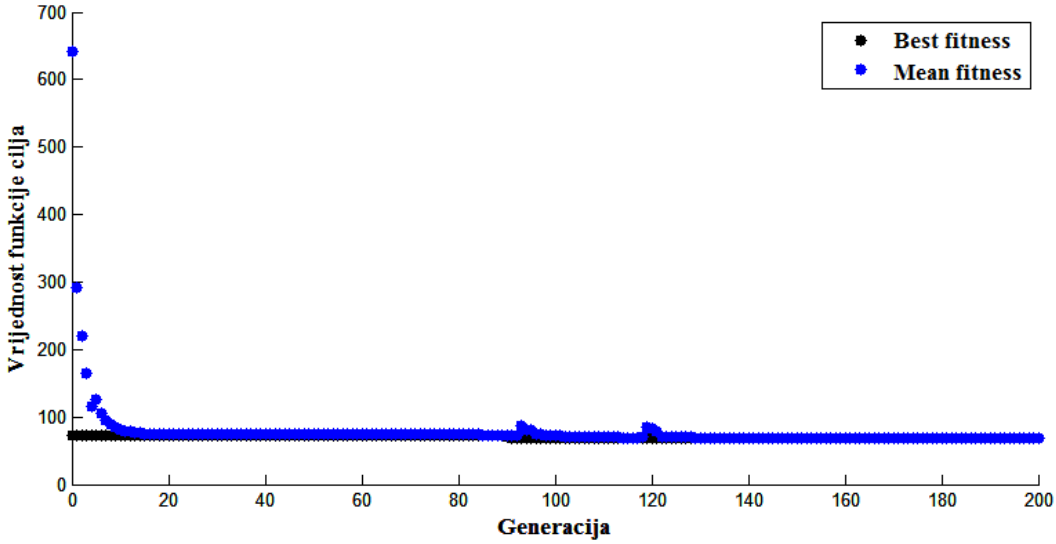
Broj Elitnih jedinki	Snaga gubitaka (kW)				Srednja vrijednost
0	72,65320	67,89293	67,88209	67,91011	69,084583
1	67,88197	67,88197	67,88197	67,88197	67,881968
2	67,88197	67,88197	67,88197	67,88197	67,881970
3	67,88197	67,88197	67,88197	68,09675	67,935664
4	67,88203	67,88199	67,88201	70,16662	68,453161

Iz tablice je vidljivo kako se najbolji rezultat dobije korištenjem jedne elitne jedinke, a vrlo blizu tome je i rješenje kod korištenja dvije elitne jedinke. Najlošije rješenje, očekivano, daje slučaj sa nula elitnih jedinki.



Slika 7.10. *Prikaz najbolje jedinke i srednje vrijednosti svih jedinki po generacijama, za slučaj sa nula elitnih jedinki*

Na slici 7.10. vidimo rezultate algoritma za slučaj kada nema elitnih jedinki. Može se primijetiti da za razliku od drugih slučajeva, ovdje najbolje rješenje može postati lošije od prethodnoga, te upravo zbog toga ovaj slučaj daje najlošije rezultate.



Slika 7.11. *Prikaz najbolje jedinke i srednje vrijednosti svih jedinki po generacijama, za slučaj sa jednom elitnom jedinkom*

U slučaju sa jednom elitnom jedinkom (Slika 7.11.) vidimo kako se iz generacije u generaciju najbolje rješenje samo poboljšava. Jedinke oko 20. generacije konvergiraju ka rješenju, no oko 100. i 120. generacije pronalaze bolja rješenja, koja su i konačna.

7.2.6. Udio križanja

U ovom se potpoglavlju ispituje utjecaj odabira udjela križanja na konačno rješenje algoritma.

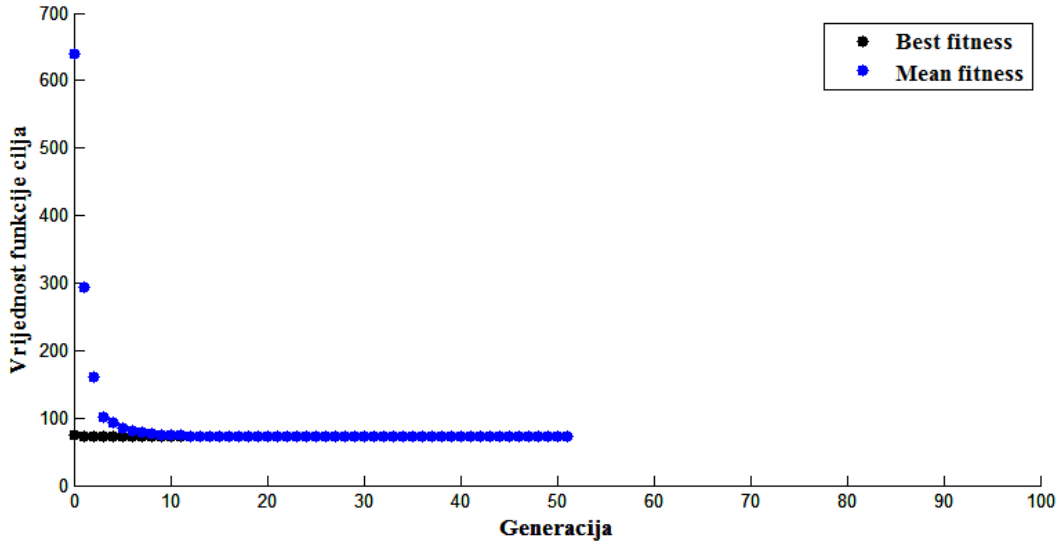
Postavke algoritma su:

- Veličina populacije: 50
- Skaliranje: S linearnim pomakom
- Funkcija selekcije: Turnirska
- Broj elitnih jedinki: 1
- Križanje: Raspršeno

Tablica 7.9. *Snaga gubitaka za različite vrijednosti udjela križanja*

Udio križanja	Snaga gubitaka (kW)				Srednja vrijednost
0	67,88197	67,88356	67,88202	67,88198	67,88238
0.2	67,88202	67,88197	67,88197	67,88197	67,88198
0.4	67,88197	67,88197	67,88197	67,88197	67,881969
0.6	67,88198	67,88197	67,88197	67,88197	67,881971
0.8	67,88197	67,88197	67,88197	67,88197	67,881968
1	72,63832	71,77819	79,59798	70,14729	73,540445

Iz tablice 7.9. možemo vidjeti kako se najbolji rezultati dobiju sa udjelom križanja od 0.8 koji je korišten i u prethodnim proračunima. Najlošiji rezultate daje udio križanja 1 (100%). Problem kod korištenja udjela križanja od 100% je što algoritam često pronađe lokalni ekstrem, umjesto globalnog.



Slika 7.12. *Prikaz najbolje jedinke i srednje vrijednosti svih jedinki po generacijama, za udio križanja 100%*

Iz slike 7.12. vidimo kako funkcija vrlo brzo konvergira, no kako se sve nove jedinke dobivaju križanjem, prostor rješenja se ne pretražuje dobro, te ovaj slučaj ne daje dobre rezultate.

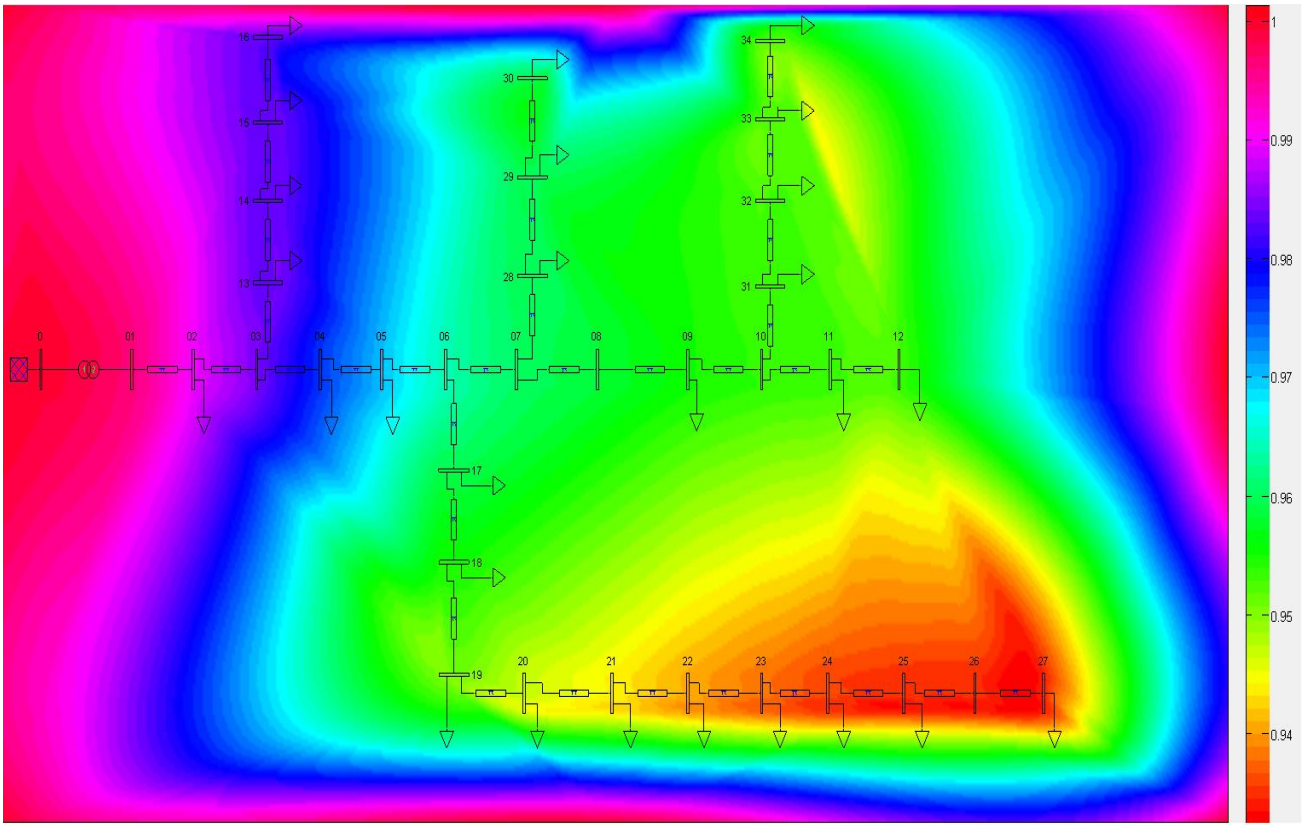
7.3. Obavljanje simulacije s konačnim postavkama algoritma

U prethodnom poglavlju napravljeni su proračuni za sve postavke, te su odabrane postavke koje daju najbolje rezultate. Konačne postavke algoritma su:

- Veličina populacije: 50
- Skaliranje: S linearnim pomakom
- Funkcija selekcije: Turnirska
- Broj elitnih jedinki: 1
- Križanje: Raspršeno
- Udio križanja: 0.8

Cilj optimiranja bio je pronaći vrijednosti snage proizvodnje distribuiranih izvora, te napon sekundara (omjer transformatora) koji za rezultat daju najmanji iznos snage gubitaka. Proračun se treba odraditi za sva opterećenja iz dnevnog dijagrama opterećenja, te za svaki od njih odrediti snage proizvodnje distribuiranih izvora, te napone sekundara transformatora.

Svi naponi u mreži ograničeni su na ±5% nazivnog napona.



Slika 7.13. *Naponska slika mreže sa slike 7.1. za slučaj kada nema distribuiranih izvora, te sa nazivnim naponom transformatora*

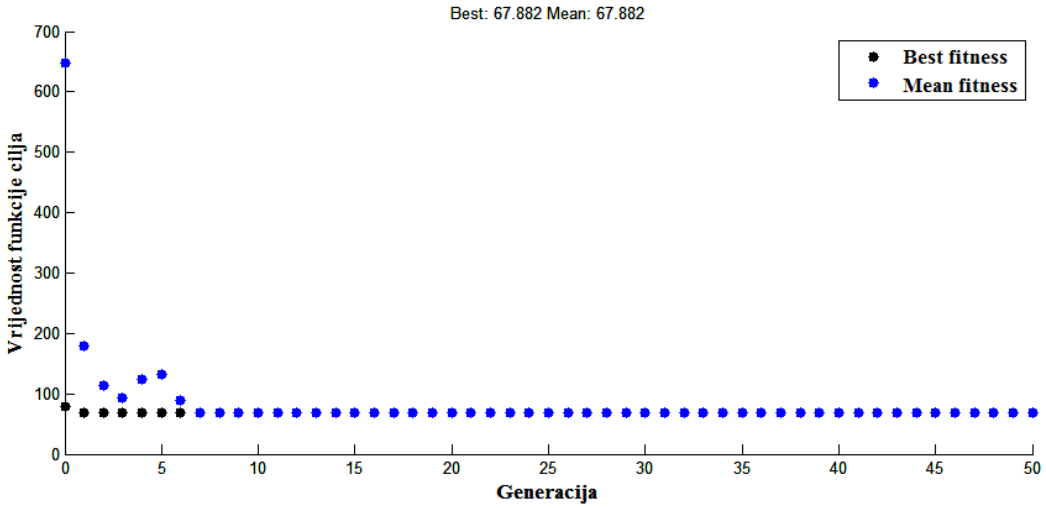
Na slici 7.13. vidimo naponsku sliku mreže koju smo koristili, prije početka optimizacije, te možemo vidjeti da je na nekim mjestima napon niži od nazivnog napona za više od 5%.

U tablici 7.10. vidimo rezultate proračuna sa konačnim postavkama algoritma za nazivno opterećenje mreže.

Tablica 7.10. *Rezultati proračuna sa konačnim postavkama algoritma*

Snaga gubitaka [kW]				Srednja vrijednost
1.	2.	3.	4.	
67,88196655	67,88196655	67,88196654	67,88196654	67,88196655

Iz rezultata se vidi kako do razlike između rješenja dolazi tek kod osme decimale, što iznosi oko 10μW. Na slici 7.14. vidimo prikaz po generacijama, te vidimo da rješenja već kod sedme generacije konvergiraju ka konačnom.



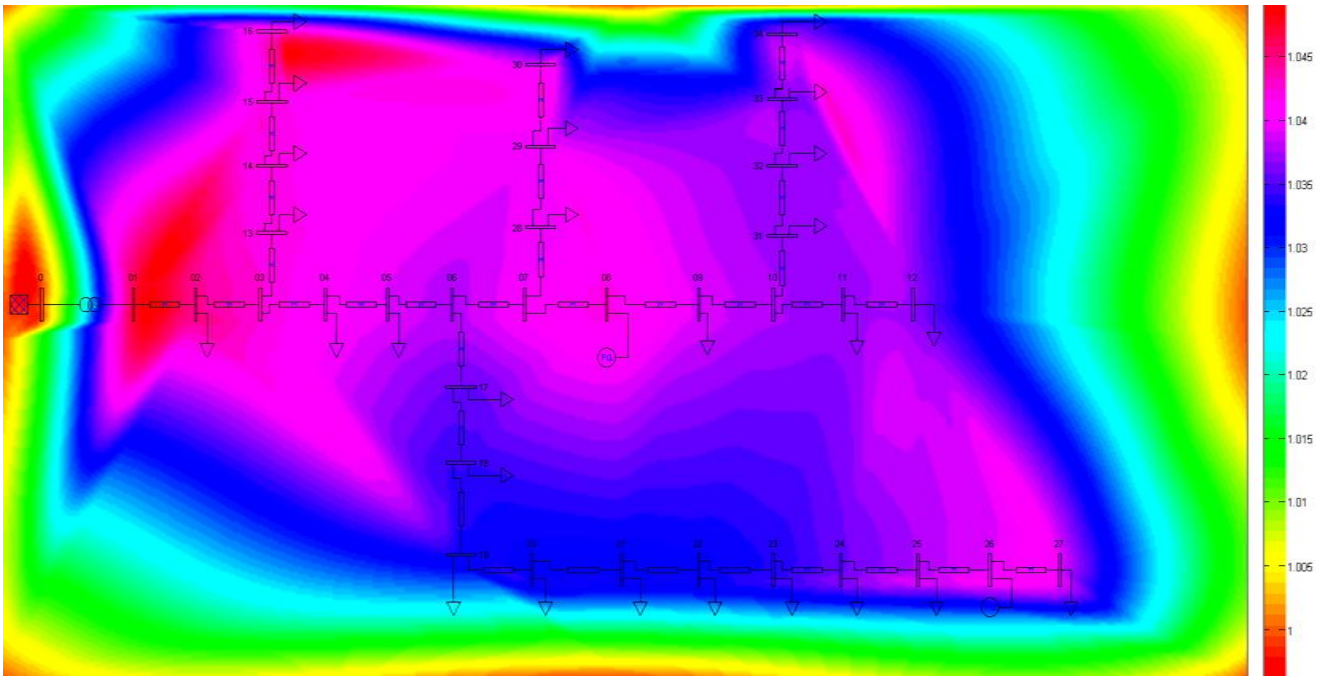
Slika 7.14. *Prikaz najbolje jedinke i srednje vrijednosti svih jedinki po generacijama za konačne postavke algoritma*

Tablica 7.11. *Iznosi proizvodnje distribuiranih izvora i napona sekundara transformatora za proračun sa konačnim postavkama algoritma*

DG1 [MW]	DG2 [MW]	Utn [kV]
1,809223	1,354328	10,5

Iznose proizvodnje distribuiranih izvora te nazivni napon sekundara transformatora vidimo u tablici 7.11. Proračun je obavljen 4 puta, te je uzeta srednja vrijednost iznosa.

Korištenjem srednje vrijednosti dobivenih rješenja, napravljena je simulacija, te je dobivena naponska slika mreže, koja se nalazi na slici 7.15.



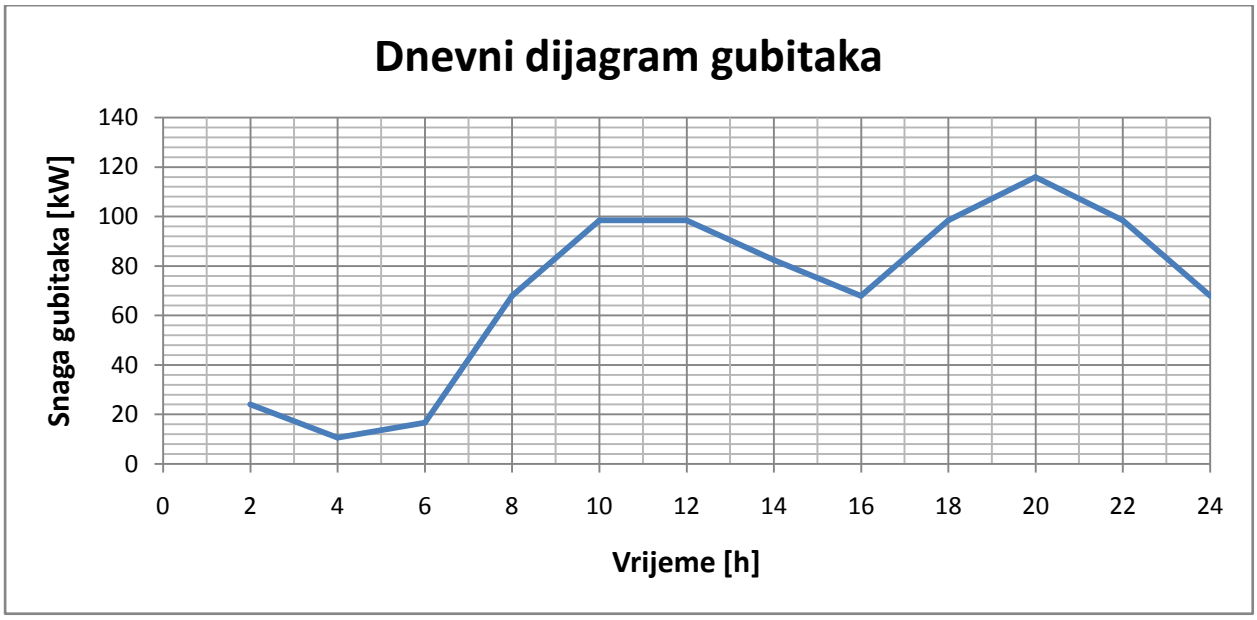
Slika 7.15. *Naponska slika optimizirane mreže*

Iz naponske slike možemo vidjeti kako naponi niti jednog dijela mreže ne prelaze ograničenje od ±5%. S obzirom da je cilj optimizacije bio smanjenje gubitaka, očekivano je da će naponi u mreži biti viši u odnosu na ne optimiziranu mrežu.

Tablica 7.12. *Iznos snage gubitaka u ovisnosti o opterećenju mreže*

Vrijeme [h]	2	4	6	8	10	12	14	16	18	20	22	24
Opterećenje [Sn]	0,6	0,4	0,5	1	1,2	1,2	1,1	1	1,2	1,3	1,2	1
Pgopt [kW]	24,1183	10,6499	16,6944	67,8819	98,4031	98,4031	82,4105	67,8819	98,4031	115,874	98,4031	67,8819
Pg [kW]	73,8885	32,1587	50,7725	214,509	316,182	316,182	262,571	214,509	316,182	375,559	262,571	214,509

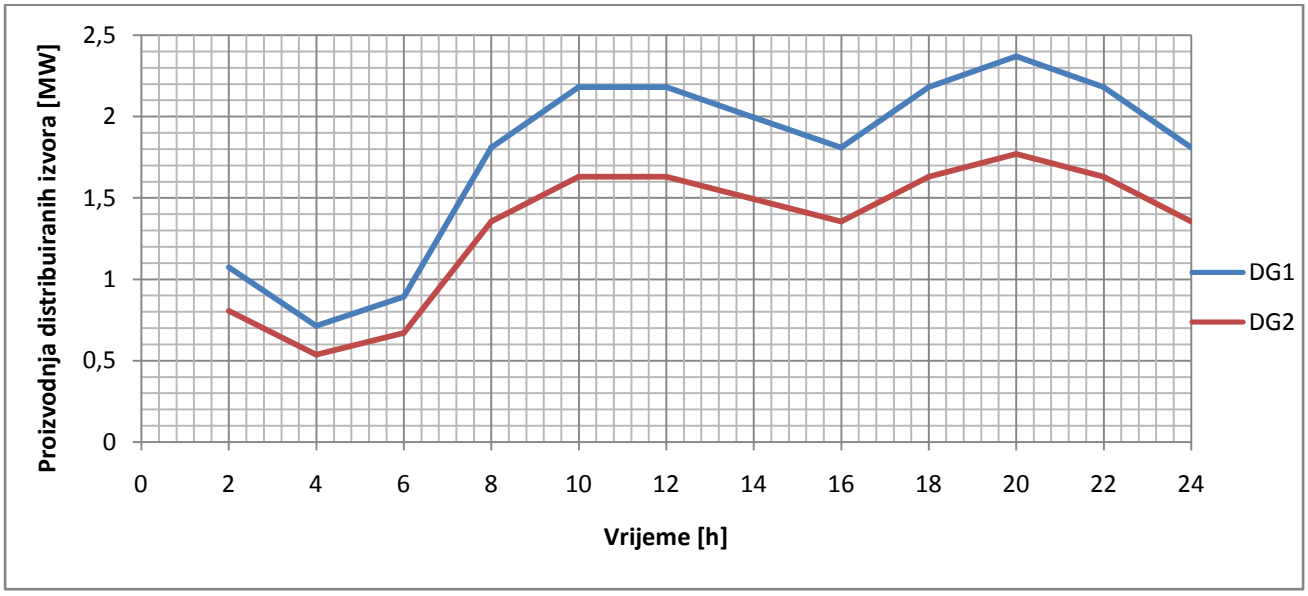
U tablici 7.12. vidimo iznose snage gubitaka za sva dnevna opterećenja, a na slici 7.16. prikazan je dijagram gubitaka, gdje se slikovito vidi promjena snage gubitaka. Kao što je i očekivano, kako raste dnevno opterećenje, tako se povećava i snaga gubitaka. U tablici se također može vidjeti usporedba gubitaka prije i poslije optimiranja, gdje se vidi koliko se zapravo smanjuju gubici. Ako izračunamo ukupne dnevne gubitke za oba slučaja, za optimiranu mrežu oni iznose oko 1694 kWh, a za mrežu koja nije optimirana iznose oko 5300 kWh, odnosno dnevno se uštedi 3606 kWh električne energije.



Slika 7.16. *Dnevni dijagram gubitaka*

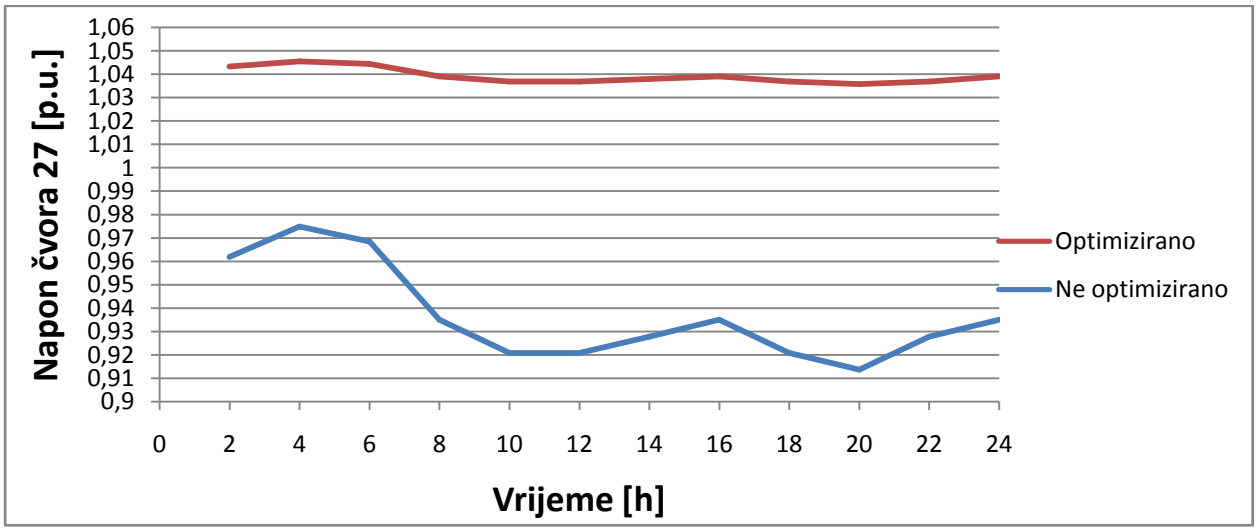
Tablica 7.13. *Promjena proizvodnje distribuiranih izvora tokom dana*

Vrijeme [h]	2	4	6	8	10	12	14	16	18	20	22	24
DG1 [MW]	1,0746	0,7128	0,8932	1,8092	2,1823	2,1823	1,9953	1,8092	2,1823	2,3704	2,1823	1,8092
DG2 [MW]	0,8070	0,5362	0,6714	1,3543	1,6308	1,6308	1,4923	1,3543	1,6308	1,7699	1,6308	1,3543

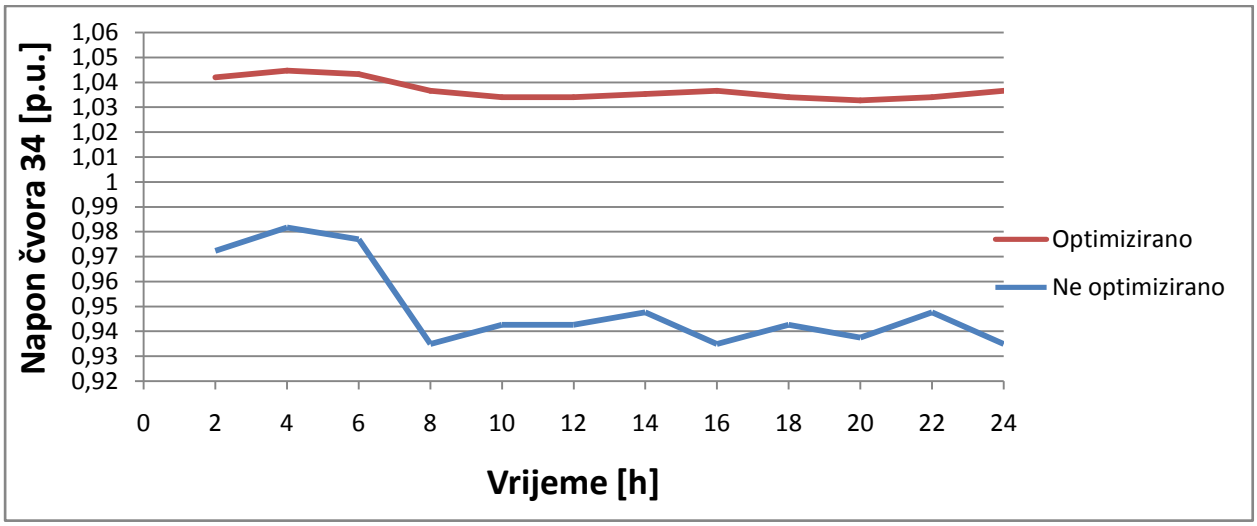


Slika 7.17. *Dnevni dijagram proizvodnje distribuiranih izvora*

U tablici 7.13. prikazani su iznosi proizvodnje distribuiranih izvora tokom dana, te su na slici 7.17 prikazani dijagramom. Može se vidjeti kako su grafovi oba distribuirana izvora vrlo sličnog oblika, odnosno da je odnos proizvodnje distribuiranih izvora, gotovo jednak tokom dana. Napon sekundara transformatora nije se mijenjao tokom dana i iznosio je 10,5 kilovolta, odnosno bio je maksimalnog iznosa.



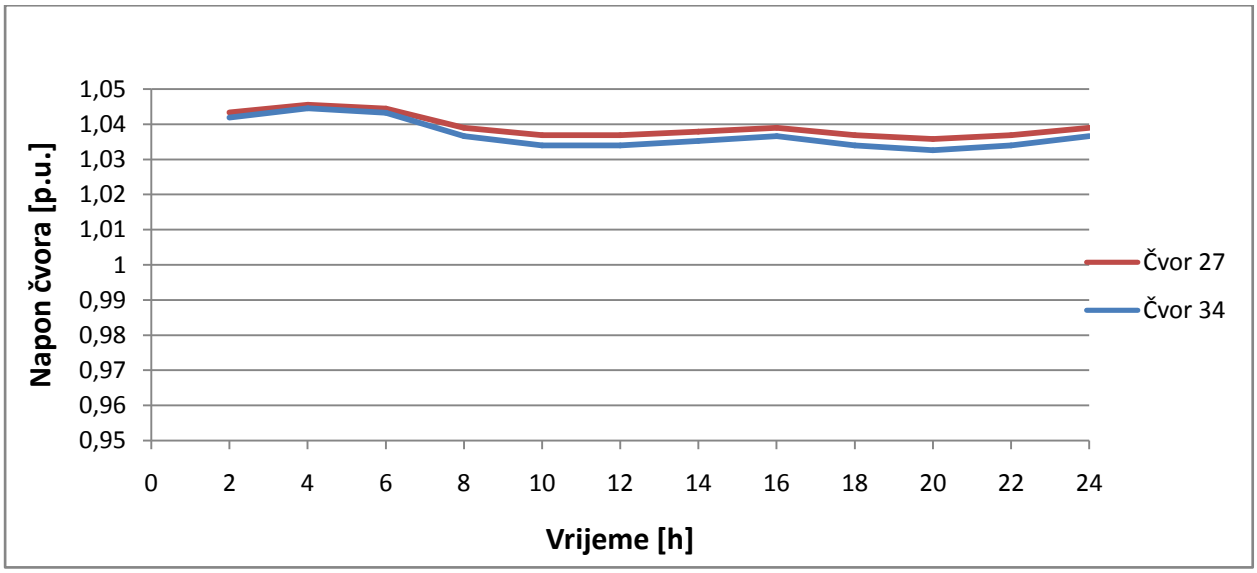
Slika 7.18. *Naponi čvora 27 za slučaj sa i bez optimizacije*



Slika 7.19. *Naponi čvora 34 za slučaj sa i bez optimizacije*

Sa slikama 7.18. i 7.19. možemo vidjeti usporedbu napona za slučaj kada je mreža optimizirana i za slučaj kada nije optimizirana. Za slučaj kada mreža nije optimizirana, pri većim

opterećenjima, napon pada ispod 95% nazivnog napona, dok je kod optimizirane mreže, tokom cijelog dana napon unutar granica ±5%.



Slika 7.20. *Naponi čvorova 27 i 34*

Na slici 7.20. možemo vidjeti kako su naponi čvora bližeg distribuiranom izvoru (čvor 27), manji od napona čvora koji je dalje od distribuiranog izvora, ali da niti jedan ne prelazi granicu od ±5% nazivnog napona.

Snaga gubitaka za nazivno opterećenja, sa ugrađenim distribuiranim izvorima sa optimalnom proizvodnjom, iznosi 67,8819 kW. Usporedimo li to sa snagom gubitaka mreže bez distribuiranih izvora i nazivnog napona transformatora, koja iznosi 214,5085 kW, možemo vidjeti kako se gubici znatno smanjuju, te da je postavljanje distribuiranih izvora u ispitanu mrežu isplativo.

8. ZAKLJUČAK

Cilj ovoga rada bio je ispitati utjecaj ugradnje distribuiranih izvora i transformatora s promjenom napona pod opterećenjem, te optimirati iznose proizvodnje i iznos napona transformatora korištenjem genetskog algoritma. Za simulaciju je korištena mreža sa 34 čvora. Za genetski algoritam korišten je gotovi alat unutar MATLAB-ovog optimizacijskog alatnog okvira. Za rad genetskog algoritma najbitnije je bilo napraviti funkciju cilja. Kod genetskog algoritma se za svaku generaciju više puta (ovisno o broju jedinki) morao odraditi proračun tokova snaga. Za proračun tokova snaga korišten je PSAT (eng. Power System Analysis Toolbox) alatni okvir koji je dodan MATALB-u, a metoda korištena za proračun tokova snaga bila je Newton-Raphsonova metoda.

U radu je također bilo potrebno odabrati postavke algoritma koje daju najbolje rezultate. Najveći utjecaj na rezultate imali su veličina populacije, funkcija selekcije i funkcija križanja, dok ostale postavke nisu toliko značajno utjecale na rezultate. Pretpostavka u ovom djelu je bila kako će odabirom pojedinačnih najboljih postavki biti dobiveni najbolji rezultati. Također kod slučaja s brojem jedinki se vidi kako se s većim brojem jedinki dobivaju bolji rezultati, no kako se povećavanjem broj jedinki znatno povećava i vrijeme izvođenja, za ostale proračune odabran je manji broj jedinki.

Iz rezultata simulacije možemo zaključiti kako je ugradnja distribuiranih izvora u mrežu korisna, te kako bi se s odgovarajućim upravljanjem znatno smanjili gubici. Međutim ugradnjom distribuiranih izvora dolaze i neki novi problemi, od kojih je najznačajniji zaštita vodova, koja je napravljena za zaštitu u samo jednom smjeru, te razni drugi manji problemi. Upravo zbog ovoga, bi se prije ugradnje distribuiranih izvora, trebala odraditi još brojna ispitivanja, kako ugradnjom ne bi došlo do kontra efekta i povećanja gubitaka, a samim time i troškova u odnosu na mrežu do tada.

9. LITERATURA

[1] Dr. sc. Nijaz Dizdarević, Dr. sc. Matislav Majstorović, Dr.sc. Srđan Žutobradić, DISTRIBUIRANA PROIZVODNJA ELEKTRIČNE ENERGIJE, Energetski institut "Hrvoje Požar", Zagreb, 2003

[2] T. Ackermann, G. Andersson, L. Söder, Distributed generation: a definition, Electric Power System Research, 57 (2001) 195-204

[3] R. Azimi, S. Esmaeili, Multiobjective daily Volt/VAr control in distribution system with distributed generation using binary ant colony optimization, Turk J Elec Eng & Comp Sci, 2013, 21: 613-629

[4] W. El-Khattam, M.M.A. Salama, Distributed generation technologies, definition adn benefits, Electric Power System Research, 71 (2004) 119-128

[5] F. A. Viawan, A Sannino, J. Daalder, Voltage control with on-load tap changers in medium voltage feedes in presence of distributed generation, Electric Power System Research, 77 (2007) 1314-1322

[6] Er. Ashis K. Mishra, Er. Y. Mohapatra, Er. Anil K. Mishra, Multi-Objective Genetic Algorithm: A Comprhensive Survey, IJETAE, Veljača, 2013.

[7] James H. Harlow, Transformer Tapchanging Under Load: A Review of Concepts and Standards, Beckwith Electric Co, Ožujak, 1993.

[8] Marin Golub, Genetski algoritam prvi i drugi dio, Fakultet elektrotehnike i računarstva, Zagreb, http://www.zemris.fer.hr/~golub/ga/ga_skripta1.pdf , http://www.zemris.fer.hr/~golub/ga/ga_skripta2.pdf (Pristup ostvaren 19.9. 2016.)

[9] M. Barukčić, Ž. Hederić, K. Miklošević, Multi Objective Optimization of Energy Production of Distributed Generation in Distribution Feeder, Faculty of Electrical Engineering of Osijek, Energycon, Svibanj 2014.

[10] Matlab R2010a, Optimization toolbox help, Quick Reference, Genetic Algorithm Solver

[11] https://hr.wikipedia.org/wiki/Evolucija (pristup ostvaren 11.6.2016.)

[12] https://hr.wikipedia.org/wiki/Deoksiribonukleinska_kiselina (pristup ostvaren 11.6.2016.)

Izvori slika:

[13] http://www.afaja.hr/solarni-paneli/ (pristup ostvaren 5.6. 2016.)

[14] https://hr.wikipedia.org/wiki/Vjetroturbine (pristup ostvaren 5.6. 2016.)

[15] https://hr.wikipedia.org/wiki/Gorivi_%C4%8Dlanak (pristup ostvaren 7.6. 2016.)

[16] https://www.researchgate.net/figure/256692920_fig2_Fig-2-Basic-concept-of-an-evolutionary-algorithm-1 (pristup ostvaren 15.6. 2016.)

[17] http://www.genetika.biol.pmf.unizg.hr/pogl15.html (pristup ostvaren 16.6. 2016.)

[18] http://www.sk.co.rs/2011/12/skpr01.html (pristup ostvaren 18.6. 2016.)

[19] http://regentsprep.org/regents/biology/units/reproduction/sexual.cfm (pristup ostvaren 24.6.2016.)

SAŽETAK

Smanjenje gubitaka električne energije i povećanje korištenja obnovljivih izvora energije postaje sve popularnije u novije vrijeme. Korištenjem distribuiranih izvora mogu se postići oba od ovih ciljeva, no potrebno je optimiranje proizvodnji distribuiranih izvora. U ovom radu korišten je genetski algoritam, koji je dio MATLAB programskog paketa. Genetski se algoritam može koristiti za rješavanje mnogih problema optimizacije. Rezultati dobiveni u radu su vrlo dobri, te pokazuju kako bi se korištenjem distribuiranih izvora mogli uvelike smanjiti gubici u mreži. Da bi se odredio stvarni utjecaj ugradnje distribuiranih izvora, potrebna su dodatna istraživanja, te uzimanje u obzir mnogih faktora koji su u radu zanemareni, kako bi se pojednostavnio rad algoritma. Stoga iako su rezultati vrlo dobri, treba ih uzeti s rezervom.

Ključne riječi: Optimiranje, distribuirana proizvodnja, distribucijski transformator, evolucija, genetski algoritam, distribucijska mreža

Control optimization of distribution transformer tap changer and power of distributed generation using evoulution lagorithms

SUMMARY

Reducing power lossee and increasing use of renewable energy resources is becoming more popular in recent times. By using distributed generation, we can achieve both of these goals, but it is necessery to optimize production of distributed generetion. In this paper we used genetic algorithm which is part of MATLAB software package. Genetic algorithm can be used for solving many optimization problems. The results obtained in this paper and they show that use of distributed generation could greatly reduce power losses in network. To determine actual impact of distributed generation usage, further research is needed, and taking into accountmany factorsthat are disregarded in this paper in order to simplify the operation of algorithm. Therefore, altough results are very good, they should be taken with reserve.

ŽIVOTOPIS

Dinko Mandarić rođen je u Virovitici 17. kolovoza 1991. godine. U Suhopolju završava osnovnu školu u "Osnovnoj školi Suhopolje". Nakon osnovne škole upisuje se u Tehničku školu Virovitica, za zanimanje Elektrotehničar. Godine 2010. završava elektrotehniču školu, te se upisuje na Elektrotehnički fakultet u Osijeku, te 2013. završava preddiplomski studij na smjeru Elektroenergetika. Iste godine upisuje se na diplomski studij Elektrotehničkog fakulteta u Osijeku, smjer Elektroenergetika, koji traje do današnjeg dana.

Prilozi

Funkcija cilja u MATLAB-u

```
function Gubici=PQ33cl(P, RAzOPTER, VcMIN,VcMAX)
omjerPSAT=0;
omjerIzGA=P(3);
if  omjerIzGA<3.3605
    omjerPSAT=3.3333;
elseif omjerIzGA>=3.3605 && omjerIzGA<3.4146
    omjerPSAT=3.3871;
elseif omjerIzGA>=3.4146 && omjerIzGA<3.4705
    omjerPSAT=3.4426;
elseif omjerIzGA>=3.4705 && omjerIzGA<3.53
    omjerPSAT=3.5000;
elseif omjerIzGA>=3.53 && omjerIzGA<3.5897
    omjerPSAT=3.5593;
elseif omjerIzGA>=3.5897 && omjerIzGA<3.6515
    omjerPSAT=3.6207;
elseif omjerIzGA>=3.6515
    omjerPSAT=3.6842;
end;


initpsat;
clpsat.mesg=0;
runpsat('MrezaPQ_mdl','data');
Settings.mva=1;
clpsat.readfile=0;
runpsat('pf');
PQ.store(:,4)=RAzOPTER*PQ.con(1:27,4);
PQ.store(:,5)=RAzOPTER*PQ.con(1:27,5);
Line.store(34,7)=omjerPSAT;
PQgen.store(1,4)=P(1);
PQgen.store(2,4)=P(2);
runpsat('pf')
Gubici=1000*Snapshot.Ploss;
Vmax=max(DAE.y(36:end));
Vmin=min(DAE.y(36:end));
if Vmax>VcMAX||Vmin<VcMIN
    Gubici=5*Gubici;
end
end
```

Kod za pozivanje genetskog algoritma:

```
tic;
% for i=2:2:24
%Genetski Algoritam
BRg=200; % broj generacija GA
BRj=50; %Broj jedinki

%Zadavanje granica varijabli
DD=[0.1 0.1 3.333];
GG=[5 5 3.6842];
```

```
BRv=size(DD,2); %Broj varijabli

%Ogranicenja napona:
VcMIN=0.95;
VcMAX=1.05;
%generiranje polazne populacije
%dodavanje krajnjih slucajeva (najvisa opterecenja, najmanja proizvodnja i
%najmanji naponi te najmanje opterecenje, najvisa proizvodnja i najvisi
%naponi)u polaznu populaciju
POPpol=[];
for j=1:BRj
    POPpol=[POPpol;DD+rand(1,size(DD,2)).*(GG-DD)];
end
%Matrica opterecenja
VrazOPTER=[2 4 6 8 10 12 14 16 18 20 22 24;...
    0.6 0.4 0.5 1 1.2 1.2 1.1 1 1.2 1.3 1.2 1];
%Odabir opterecenja iz krivulje opterecenja
%Sat=20; %Odabir perioda vremena u krivulji opterecenja
% for i=2:2:24
     Sat=8;
RAzOPTER=[];
for k=1:size(VrazOPTER,2)
    if VrazOPTER(1,k)==Sat
     RAzOPTER=VrazOPTER(2,k);
    end
end


pos=gaoptimset('Generations',BRg,'InitialPopulation',POPpol,'PopulationSize'.
..
    ,BRj,'Vectorized','off','UseParallel','always'...
    ,'SelectionFcn',{@selectiontournament, 4},'CrossoverFcn',...
    {@crossoverscattered},'EliteCount',1,'TolFun',1E-12,'CrossoverFraction',
0.8,'FitnessScalingFcn',@fitscalingshiftlinear,'MutationFcn',@mutationadaptfe
asible, 'PlotFcns',{@gaplotbestf});
%{@selectiontournament,2} @crossoverscattered 'InitialPopulation',POPpol,
%,@gaplotbestindiv
OFV=[];
%
[RJ,OFV,EXf,IZL,POPrj]=ga(@(P)
PQ33cl(P,RAzOPTER,VcMIN,VcMAX),BRv,[],[],[],[],DD,GG,[],pos);


TAPtrafo=RJ(3);
if  TAPtrafo<3.3605
    RJ(3)=3.3333;
elseif TAPtrafo>=3.3605 && TAPtrafo<3.4146
    RJ(3)=3.3871;
elseif TAPtrafo>=3.4146 && TAPtrafo<3.4705
    RJ(3)=3.4426;
elseif TAPtrafo>=3.4705 && TAPtrafo<3.53
    RJ(3)=3.5000;
elseif TAPtrafo>=3.53 && TAPtrafo<3.5897
    RJ(3)=3.5593;
elseif TAPtrafo>=3.5897 && TAPtrafo<3.6515
    RJ(3)=3.6207;
elseif TAPtrafo>=3.6515
    RJ(3)=3.6842;
end;
```

```
initpsat;
clpsat.mesg=0;
runpsat('MrezaPQ_mdl','data');
Settings.mva=1;
clpsat.readfile=0;
runpsat('pf');
PQ.store(:,4)=RAzOPTER*PQ.con(1:27,4);
PQ.store(:,5)=RAzOPTER*PQ.con(1:27,5);
Line.store(34,7)=RJ(3);
PQgen.store(1,4)=RJ(1);
PQgen.store(2,4)=RJ(2);
runpsat('pf')

toc;
beep
```