Primjena metoda optimizacije u elektroenergetskom sustavu

Palčić, Hrvoje

Master's thesis / Diplomski rad

2018

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:200:386527

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

PRIMJENA METODA OPTIMIZACIJE U ELEKTROENERGETSKOM SUSTAVU

Diplomski rad

Hrvoje Palčić

Osijek, 2018. godina

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Obrazac D1: Obrazac za imenovanje Povjerenstva za obranu diplomskog rada

Osijek, 26.03.2018.

Odboru za završne i diplomske ispite

Imenovanje Povjerenstva za obranu diplomskog rada

Ime i prezime studenta:	Hrvoje Palčić
Studij, smjer:	Diplomski sveučilišni studij Elektrotehnika
Mat. br. studenta, godina upisa:	D 955, 29.09.2017.
OIB studenta:	31417017656
Mentor:	Izv. prof. dr. sc. Predrag Marić
Sumentor:	Marko Vukobratović
Sumentor iz tvrtke:	
Predsjednik Povjerenstva:	Doc.dr.sc. Marinko Barukčić
Član Povjerenstva:	Marko Vukobratović
Naslov diplomskog rada:	Primjena metoda optimizacije u elektroenergetskom sustavu
Zadatak diplomskog rada:	Prikazati pregled metoda optimizacije koje imaju primjenu u elektroenergetskom sustavu, a na jednostavnijem primjeru elektroenergetske mreže prikazati primjenu (primjerice optimalan smještaj uređaja za kompenzaciju , optimalan smještaj distribuirane proizvodnje..) jedne od tih metoda.
Prijedlog ocjene pismenog dijela ispita (diplomskog rada):	Dobar (3)
Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:	Primjena znanja stečenih na fakultetu: 2 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 2 bod/boda Jasnoća pismenog izražavanja: 3 bod/boda Razina samostalnosti: 1 razina
Datum prijedloga ocjene mentora:	26.03.2018.

Potpis mentora za predaju konačne verzije rada u Studentsku službu pri završetku studija:	Potpis:
	Datum:

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK**

IZJAVA O ORIGINALNOSTI RADA

Osijek, 18.06.2018.

Ime i prezime studenta:	Hrvoje Palčić
Studij:	Diplomski sveučilišni studij Elektrotehnika
Mat. br. studenta, godina upisa:	D 955, 29.09.2017.
Ephorus podudaranje [%]:	11%

Ovom izjavom izjavljujem da je rad pod nazivom**: Primjena metoda optimizacije u elektroenergetskom sustavu**

izrađen pod vodstvom mentora Izv. prof. dr. sc. Predrag Marić

i sumentora Marko Vukobratović

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija.
Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis studenta:

SADRŽAJ

1. UVOD 1
2. OPTIMIZACIJA OPĆENITO 2
2.1. Matematička definicija optimizacije 2
2.2. Metode optimizacije 6
2.2.1. Konvencionalne metode 8
2.2.2. Metode inteligentnog pretraživanja 12
3. OPTIMALNA RASPODJELA SNAGE I ANGAŽIRANOST PROIZVODNIH JEDINICA 16
3.1. Pretpostavke mreže 16
3.2. Pravila za stvaranje probnog rješenja 17
3.3. Problemi optimalne raspodjele 17
3.4. Troškovi proizvodnje 18
3.5. Troškovi pokretanja 18
3.6. Ograničenja 19
3.7. Pravila za generiranje probnih rješenja 21
3.8. Generiranje inicijalnog rješenja 23
4. METODA ROJA ČESTICA 25
4.1. Opis algoritma 26
4.2. Generiranje i inicijalizacija algoritma 26
5. SIMULACIJE 29
5.1. Metoda roja čestica (PSO) 29
5.2. Optimizacija pomoću Newton-Raphson i Gauss- Seidel metode 34
5.2.1. Rezultati za Newton-Raphson metodu 36
5.2.2. Rezultati za Gauss- Seidel metodu 38
6. ZAKLJUČAK 41
7. LITERATURA 42
SAŽETAK 44
SUMMARY 45
ŽIVOTOPIS 46

1. UVOD

Sve veći napredak tehnologija, kao i obnovljivih izvora energije uzrokovao je veliko širenje distribuiranih izvora u elektroenergetskoj mreži. Tehnologija obnovljivih izvora postala je financijski isplativa te se sve više investira u njih kako bi se smanjili troškove ili zarađivalo od njih. Također svijest oko globalnog zatopljenja i ekologije je porasla, a time su dani i poticaji za obnovljive izvore energije što je bio dodatni razlog naglog rasta distribuiranih izvora na mreži. Kako je elektroenergetski sustav sam po sebi bio složen, dodavanjem mnogih malih izvora mreža se dodatno zakomplicirala, u odnosu na kada je mreža bila centralizirana i kada su tokovi snaga bili jednosmjerni. Zbog toga je potrebno predvidjeti utjecaj novih izvora na postojeću mrežu. Zbog novih izvora tokovi snaga se mijenjaju, ponekad novi izvori povećavaju gubitke u mreži te je potrebno mijenjati i zaštitu koja je do distribuiranih izvora bila jednosmjerna.

Cilj ovog rada je prikazati metode koje služe za optimizaciju mreže, koje imaju primjenu u elektroenergetskom sustavu. Na primjeru elektroenergetske mreže prikazati će se primjena jedne od metoda. Za predviđanje gubitaka i utjecaja na mrežu koriste se razni algoritmi, u svrhu pronalaska optimalnog podešenja promatrane mreže. Metode koje će se koristiti su metoda roja čestica (engl. Particle Swarm Optimisation, PSO), stohastični algoritam čiji se rad bazira na ponašanju jata ili roja životinja u prirodi. Za izvođenje simulacija korišten je programski paket *MATLAB* (engl. *MATrixLABoratory)* i njegova nadogradnja MATPOWER. U radu se pomoću simulacije navedenom metodom optimira mreža kako bi se dobili najmanji gubici tokom prijenosa energije.

2. OPTIMIZACIJA OPĆENITO

Optimizacija se izvršava kako bi se pronašlo optimalno rješenje nekog problema, a promatrani problemi mogu posjedovati više parametara koji utječu na samo rješenje. Prema tome optimizacijom nekog problema pronalazi se kombinacija parametara koji nam najbolje odgovaraju. Stoga se u konačnici često dobiva veći broj mogućih alternativa rješenja. Iz cjelokupne skupine mogućih rješenja nastoji se pronaći najbolju ili u ovisnosti o prihvaćenim uvjetima optimalnu alternativu. Naime, optimizacija je složen problem povezan sa višestrukim ciljevima, nesigurnostima i većim brojem promjenjivih veličina. Obzirom da se radi o složenom problemu dolazi se do više ciljnog optimizacijskog zadatka [2].
Proces planiranja sastoji se od nekoliko koraka temeljem kojih se sukladno zadanim kriterijima, dobiva niz pogodnih rješenja. [3]:

1. Identificiranje problema – definiranje opsega primjene i ograničenja;
2. Određivanje ciljeva – definiranje što treba minimizirati;
3. Identificiranje rješenja – koje su dostupne opcije;
4. Proračun rješenja– proračun svih mogućnosti;
5. Odabir najbolje rješenja izbor opcije koja najbolje zadovoljava ciljeve;
6. Donošenje konačne odluke – temeljem dobivenih rezultata odabir alternative koja će se u konačnici i provesti.

2.1. Matematička definicija optimizacije

Optimizacija je u konačnici matematička funkcija kojom se rješava jedan ili više problema.
Ako se rješava problem koji treba zadovoljiti više zahtjeva tada je to višeciljni optimizacijski zadatak. Rješenje funkcije će biti višestruko, odabrat će se rješenje koje je optimalno tj. najbolje. Zbog mogućnosti računanja ishoda s višestrukim ciljevima metode optimizacije su vrlo popularne u znanosti, raznim eksperimentima, kao i za rješavanje problema optimizacije elektroenergetske mreže. Optimiranjanje se matematički može objasniti kao pronalaženje ekstrema zadane funkcije, jer za cilj kod optimiranja većinom trebamo pronaći maksimum ili minimum funkcije.

Matematički zapis ovisno o tome tražimo li minimum ili maksimum funkcije cilja je :

$$\min f(x_i, x_2, x_3, x_4, \ldots, x_n) \tag{2-1}$$

Ili za maksimum:

$$\max f(x_i, x_2, x_3, x_4, \ldots, x_n) \tag{2-2}$$

Funkcija cilja je $(x_i, x_2, x_3, x_4, \ldots, x_n)$ koja ovisi o n varijabli. Svaka varijabla može biti ograničena, a ograničenja su eksplicitna i implicitna. Varijable koje se zadaju gornjom i donjom vrijednosti varijable nazivaju se implicitna, te ograničenja koja su eksplicitna dijele se na ograničenja jednadžbi i nejednadžbi [12].

$$h_i(x) = 0, i = 1, \ldots, m \tag{2-3}$$

$$g_j(x) \leq 0, \; j = 1, \ldots, k \tag{2-4}$$

Kako bi se funkcija ograničenja zamijenila sa funkcijom bez ograničenja koristit će se metoda Lagrange-ovih množitelja. Nakon što se ova metoda primjeni na funkciju, tada je funkciju moguće zapisati izrazom (2-5).

$$L(x, \lambda, \mu) = f(x) + \sum_{i=1}^{m} \lambda_i h_i(x) + \sum_{j=1}^{1} \mu_j g_j \tag{2-5}$$

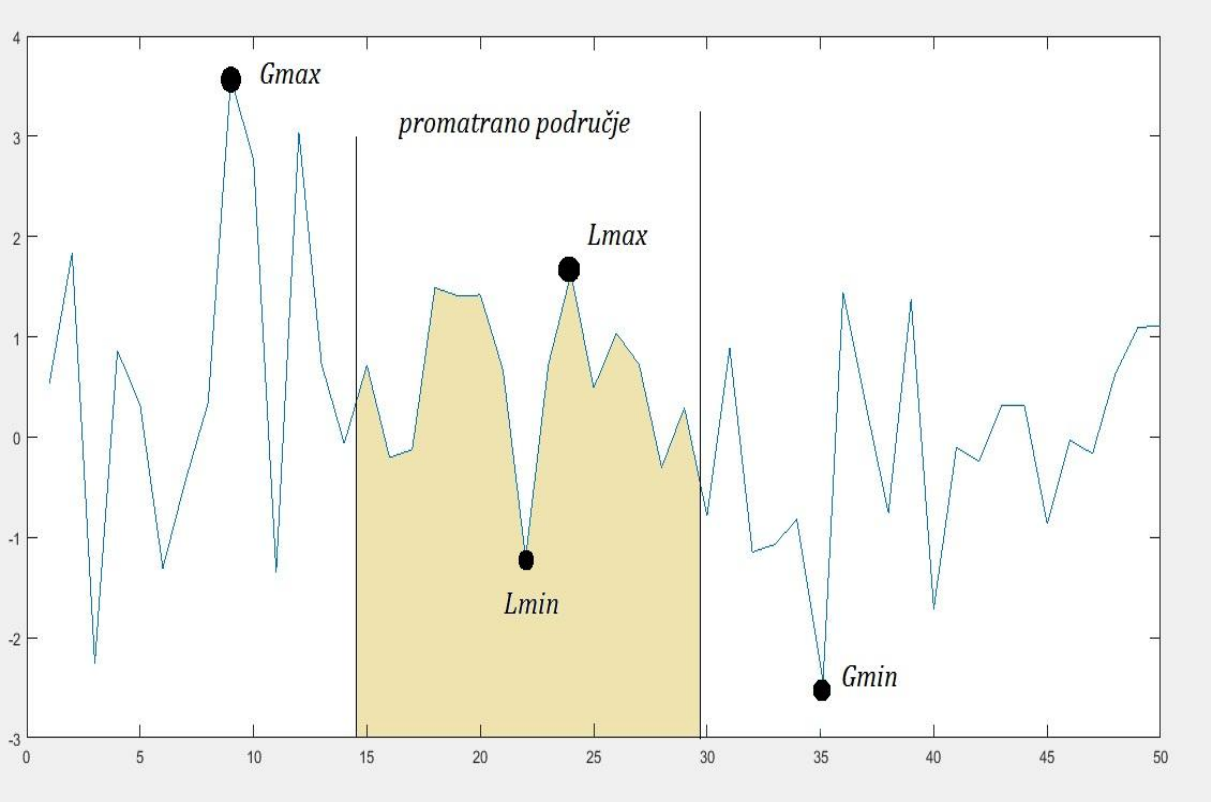
h_i - j-to ograničenje jednakosti.

g_j - i-to ograničenje jednakosti

μ_j - konstanta kojom se množi j-to ograničenje jednakosti

λ_i - konstanta kojom se množi i-to ograničenje jednakosti

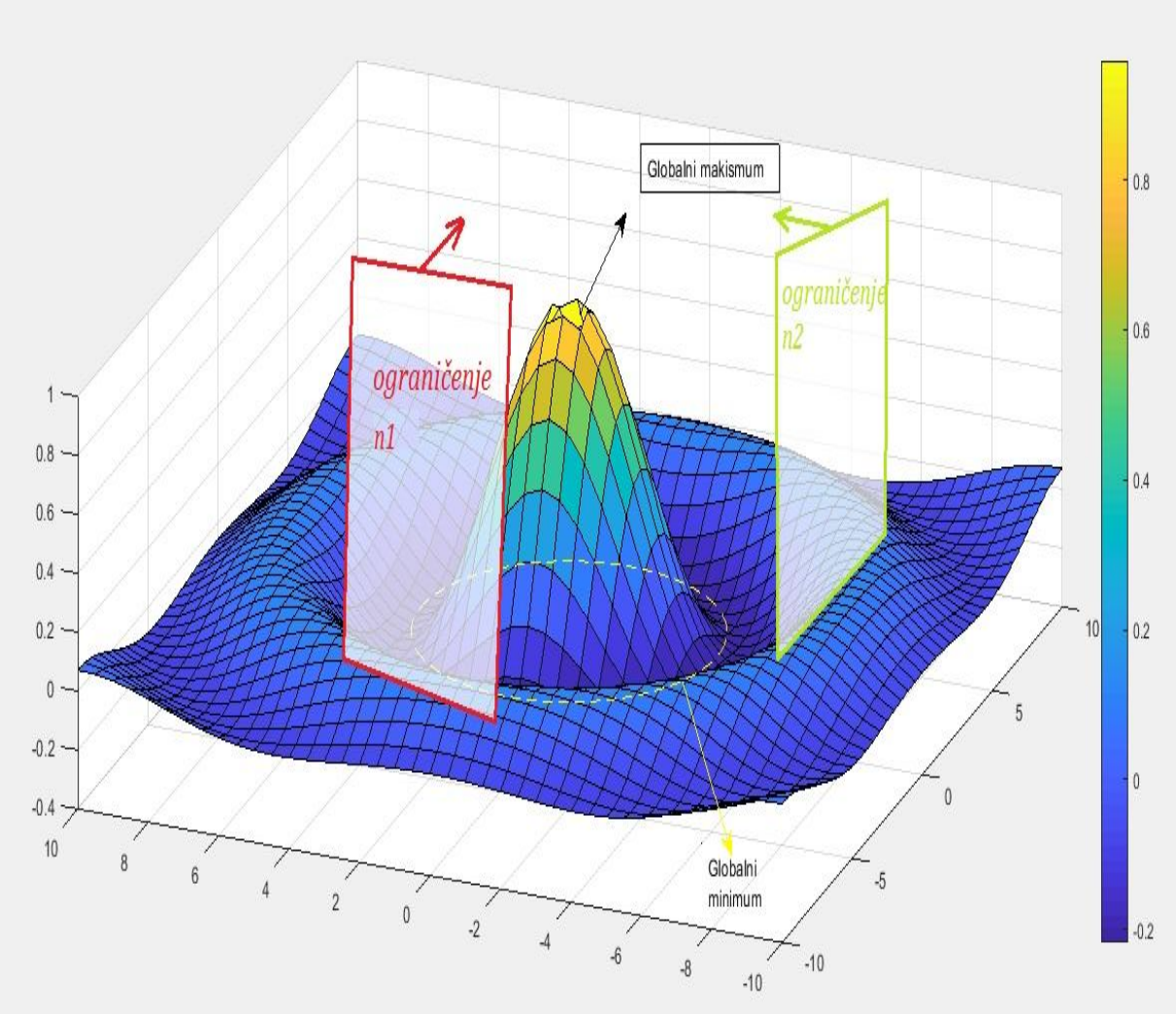
Rješenja funkcije općenito imaju više rješenja koja se nazivaju lokalni ekstremi, a globalni ekstremom se smatra najmanja ili najveća vrijednost funkcije. Optimiranjem pronalazimo globalni ekstrem zadane funkcije uz zadovoljenja svih zadanih ograničenja varijabli.
Slikom 2.1. prikazano je da globalni ekstremi ne moraju biti rješenja zadane funkcije cilja jer ne spadaju u zadana ograničenja, stoga je češći slučaj da su lokalni ekstremi rješenja problema optimiranja nekog procesa.



Slika 2.1. *Ekstremi funkcije*

Ciljevi optimiranja mogu biti raznovrsne prirode, kao što su smanjenje gubitaka, povećanje efikasnosti, smanjenje troškova, smanjenje emisije štetnih plinova, itd., takvi slučajevi optimiranja nazivaju se jednociljni. Čest slučaj kod optimiranja je da je je cilj višestruk ili višeciljni, te se tada ponekad događa da su ciljevi kontradiktorni, na primjer želi se povećati proizvodnja i smanjiti troškovi. Logično je da u tom slučaju neće biti moguće jedno najbolje rješenje, već će se stvoriti kompromis između željenih ciljeva. U prethodnom tekstu je

naveden proces planiranja kada se susreće s višeciljni optimizacijskim zadatak. Na slici 2.2. može se vidjeti grafički prikaz kompleksnije funkcije cilja s višeciljnim ograničenjima.



Slika 2.2. *Prikaz kompleksne funkcije s ograničenjima*

Slikom 2.2. uočava se kako ograničenja djeluju na ishod rješenja. Zbog ograničenja n1 i n2, prostor rješenja se suzio, čime se za rješenje dobivaju lokalni ekstremi koji spadaju u ograničeno područje.

2.2. Metode optimizacije

Metode optimizacije se zasnivaju na optimizacijskim algoritmima koji su prikladni za određenu vrstu problema. Vrlo je važno dobro identificirati problem kako bi se pronašlo najbolje rješenje problema tj. najbolji algoritam za rješavanje specifičnog problema. Svaki problem ima određene zahtjeve maksimizacije, minimizacije, različite računalne zahtjeve, matematičke zahtjeve i tako dalje. Problemi se razvrstavaju prema matematičkim značajkama funkcije cilja, ograničenja i kontrolne varijable.
Postoje dvije osnovne klase optimizacijskih metoda prema vrsti rješenja [1].

1) Kriteriji optimalnosti

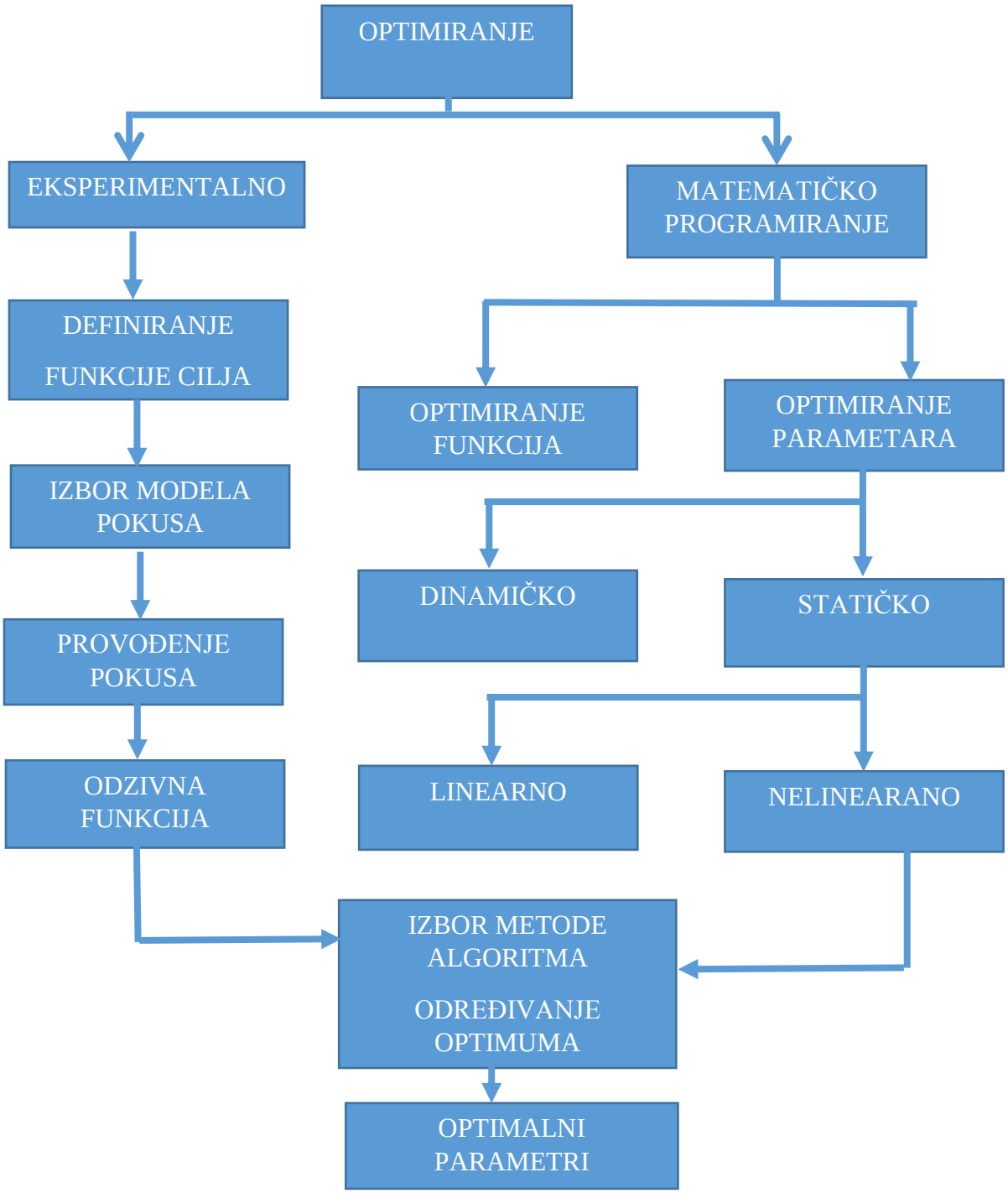
   Analitičke metode se koriste kada se utvrde uvjeti za optimalno rješenje.
   Tada se testiranjem provjerava da li kandidati zadovoljavaju zadane uvjete.
   Iz kriterija optimalnosti izvede se jednadžba kako bi se analitički odredilo optimalno rješenje.

2) Načini pretraživanja

   Korištenjem numeričke metode odabire se inicijalno probno rješenje, bilo pomoću korištenja logike ili slučajno, te se procjenjuje funkcija cilja. Tada se kreće na novu točku (drugo probno rješenje) i funkcija cilja se ponovno vrednuje. Ako je novo rješenje manje od vrijednosti za prvo probno rješenje, vrijednost se zadržava i pokreće se novi korak. Proces se ponavlja sve dok se ne pronađe minimum.

   Metode pretraživanja se koriste kada je:

   - Broj varijabli i ograničenja je velik.
   - Funkcija problema (cilj i ograničenje) je vrlo nelinearna.
   - Funkcija problema (cilj i ograničenje) je implicitna, u smislu odluka i kontrolnih varijabli koje otežavaju dobivanje informacija.



Slika 2.3. *Blokovski prikaz osnovnih podjela optimiranja (preuzeto iz [15])*

Optimiranje se grubo može podijeliti na eksperimentalno i matematičko programiranje [1]. Eksperimentalno optimiranje se zasniva na statističkim podacima, ili na podacima dugotrajnih mjerenja, služi kako bi se vrlo kompliciran proces optimirao bez izrade vrlo kompleksnog matematičkog modela, koji bi u konačnici bio i vrlo netočan. U ovome radu baza daljnjih promatranja je matematičko programiranje.

Ostali prijedlozi za klasifikaciju metoda optimizacije su [1]:

1. Prva metoda optimizacije se temelji na klasičnim metodama kao što je tehnika nelinearnog programiranja, metoda utega i metode ε-ograničenja.

2. Druga metoda optimizacije se temelji na evolucijskim tehnikama kao što je NPGA metoda (Niched Pareto genetski algoritam), NSGA (*engl.Non-dominated Sorting Genetic Algorithm*), SPEA (*engl. Strength Pareto Evolutionary Algorithm*) i SPEA2 (*engl. Strength Pareto Evolutionary Algorithm 2*).

Optimizacijske metode mogu se uglavnom podijeliti u tri skupine:
a) *Konvencionalne optimizacijske metode*;
b) *Metode inteligentnog pretraživanja*;
c) *Metode rješavanja neizvjesnosti ciljeva i ograničenja*.

2.2.1. Konvencionalne metode

Konvencionalne metode posjeduju određena ograničenja, izazove i nesigurnosti kod izvođenja optimizacije. Neka od njih su dugo vrijeme izvođenja optimizacije, jako dobro poznavanje problema i varijabli koje utječu na ishod. Također klasične metode su vrlo kompliciranе što dodatno otežava dobivanja optimalnog rješenja. Svi ovi čimbenici čine konvencionalne metode neprimjenjivima za mnoge probleme, a pogotovo primjene u elektroenergetskom sustavu. U povijesti su se ovim metodama koristili neki od poznatijih znanstvenika Lagrange, Fourier, Newton, Gauss itd., te su se kasnije ove metode unaprjeđivale i postale temelj daljnjeg razvitka algoritama i metoda optimizacije.

Skupinu konvencionalnih metoda i algoritama uključuje[2]:

1) optimizaciju bez ograničenja (*engl. unconstrained optimization*),
2) linearno programiranje (*engl. linear programming*),
3) nelinearno programiranje (*engl. nonlinear programming*),
4) kvadratno programiranje (*engl. quadratic programming*),
5) Newtonovu metodu (*engl. Newton method*),
6) metodu unutarnje točke (*engl. Interior point method*),
7) mješovito‑cjelobrojno programiranje (*engl. mixed‑integer programming*),
8) programiranje mrežnih tokova (*eng. network flow programming*).

1) Optimizacija bez ograničenja

Optimizacijski pristup bez ograničenja je temelj optimizacijskih algoritama sa ograničenjima. Većina optimizacijskih problema sa ograničenjima, u distribucijskom elektroenergetskom sustavu, mogu se prevesti u optimizacijske probleme bez ograničenja. Glavne optimizacijske metode bez ograničenja, koje se koriste u optimizaciji distribucijskih elektroenergetskih sustava, su: gradijentna metoda (*engl. gradient method*), linijsko pretraživanje (*engl. line search*), Lagrangeova metoda multipliciranja (*engl. Lagrange multiplier method*), Newton-Raphsonova metoda (*engl. Newton‑Raphson method*), kvazi‑Newtonova metoda (*engl. quasi‑Newton method*) itd [2].

2) Linearno programiranje

Linearno programiranje je pouzdana tehnika za rješavanje velikog broja optimizacijskih problema s linearnim ograničenjima i ciljevima. Programski paketi za optimizaciju elektroenergetskih sustava koji se koriste za linearizaciju nelinearnih optimizacijskih problema distribucijskog elektroenergetskog sustava, sadrže snažne algoritme linearnog programiranja [2]. Linearno programiranje koristi simpleks metodu, revidiranu simpleks metodu i tehnike unutarnje točke za rješavanje linearnih problema. Najpoznatija je simpleks metoda (*engl. simplex method*). U praksi su rezultati u inženjerskom smislu zadovoljavajući, u rješavanju problema distribucijskog elektroenergetskog sustava kao što su optimizacija tokova snage i jalove snage, upravljanje elektroenergetskim sustavom itd.

Ovaj pristup ima nekoliko prednosti, od kojih su neke [2]:

- Pouzdanost, naročito kada su u pitanju svojstva konvergencije;
- Brzo identificiranje neizvodljivosti;
- Pribrajanje velikog izbora ograničenja distribucijskog elektroenergetskog sustava, uključujući veoma važna nepredviđena ograničenja.

Nedostaci ovog pristupa su:

- Netočna procjena gubitaka sustava;
- Nedovoljna preciznost u pronalaženju točnog rješenja.

3) Nelinearno programiranje

Ova metoda se koristi za nelinearne probleme, kao što su problemi u distribucijskom sustavu, kod kojih su funkcija cilja i ograničenja nelinearna. Prvi korak u primjeni ovih tehnika je odabir smjera pretraživanja u iterativnoj proceduri, koji je određen prvom parcijalnom derivacijom jednadžbi (reducirani gradijent)[2]. Zato se ove tehnike mogu klasificirati kao metode prvog reda. Metode nelinearnog programiranja imaju znatno veću točnost nego metode linearnog programiranja i također imaju globalnu konvergenciju, što znači da ovakva konvergencija garantira rješenje koje je neovisno o polaznoj točki. Ipak, u određenim slučajevima konvergiranje prema rješenju može biti sporo radi cik-cak kretanja u prostoru pretraživanja [2].

4) Kvadratno programiranje

Kvadratno programiranje je metoda za rješavanje nelinearnih ograničenih problema. Ova metoda se bazira na generalizaciji Newtonove metode za neograničene probleme minimizacije. Funkcija cilja je kvadratna dok ograničenja imaju linearnu formu. Ova metoda je točnija od linearnog programiranja, te se u praksi koristi za optimizaciju cijena troškova generatora, troškovi su funkcija cilja koja je u pravilu kvadratna [2].

5) Newtonova metoda

Newtonova metoda ima zahtjeve računanja parcijalnih derivacija drugog reda u jednadžbama tokova snaga u kojima postoje određena ograničenjima, također se još naziva i metodom drugog reda. Nužni uvjeti optimalnosti uobičajeno su Kuhn‑Tuckerovi uvjeti. Newtonova metoda je u širokoj uporabi zbog svojih kvadratnih svojstava konvergencije [2].

6) Metoda unutarnje točke

Prema autoru Frani Tomaševiću, metoda unutarnje točke razvila se sa ciljem rješavanja linearnog programiranja. Ova metoda posjeduje više prednosti naspram konvencionalne simpleks metode linearnog programiranja, prednost ove metode je brzina računanja. Koristi se za računanje optimalnih tokova snaga i optimalne raspodjele jalove snage [3].

7) Metoda mješovitog cjelobrojnog programiranja

Metoda mješovitog cjelobrojnog programiranja se temelje na pretpostavci da su optimizacijski problemi mješovito cjelobrojni, te da su varijable problema cjelobrojne. U elektroenergetskom sustavu neke od tih cjelobrojnih varijabli su kut faznog pomaka, prijenosni omjer transformatora, status uključenosti nekih od elemenata i slično. Ova metoda je vrlo računalno zahtjevna, te se zbog toga dugo izvodi jer posjeduje velik broj diskretnih varijabli. Da bi se proces optimizacije ubrzao problem se razdvoji na kontinuirane i na cjelobrojne probleme pomoću dekompozicijskih tehnika kao što je npr. Bendersova dekompozicija. Rezultati dekompozicije bi značajno trebali smanjiti broj iteracija, kao i vrijeme izvođenja proračuna. Ova se metoda može koristit za rješavanje optimalnih tokova snaga, određivanje voznog reda, optimalnu rekonfiguraciju distribucijske mreže.

8) Programiranje mrežnih tokova

Ova metoda je specijalni oblik linearnog programiranja. Prvenstveno je korištena na linearnim modelima, a kasnije se za optimizaciju mrežnih tokova elektroenergetskog sustava koristi

metoda nelinearnog konveksnog programiranja. Programiranje mrežnih tokova karakterizira jednostavnost i brzina izvođenja algoritama [2].

2.2.2. Metode inteligentnog pretraživanja

Metode inteligentnog pretraživanja pripadaju heurističkim (iskustvenim) metodama i danas imaju primjenu u rješavanju širokog spektra optimizacijskih problema. Ova metoda je dobar odabir za rješavanje problema koji se teško matematički formuliraju. Također, prigodne su za rješavanje problema koji bi se inače mogli rješavati raznim kompliciranijim matematičkim metodama, međutim zbog nedostatka ili nepouzdanosti raspoloživih podataka nisu prigodne. Ove metode su prigodne za rješavanje problema sa poznatim matematičkim algoritmom, ponekad čak i jednostavnim, ali bi njegova primjena zbog velikog broja nepoznatih veličina zahtijevala nedopustivo mnogo vremena[2] . Metode inteligentnog pretraživanja tj. umjetne inteligencije nastoje izbjeći rješenja koja vrlo vjerojatno nisu ekstrem funkcije, te se može zaključiti da tada niti ne sadrže optimum. Ovim načinom rješenja se dobivaju relativno velikom brzinom, ali se teško može sa sigurnošću reći da je to rješenje najbolje, stoga se to rješenje ne može niti smatrati optimalnim rješenjem jer to ne možemo dokazati. Za ovakva rješenja se može jedino tvrditi, sa velikom vjerojatnošću, da su ona bolja od rješenja koja bi se dobila bez primjene određene metode. Lista ovih metoda je dosta dugačka, ali neke od najšire upotrebljavanih pri rješavanju optimizacijskih problema su [2]:

1) Monte Carlo metoda (*engl. Monte Carlo method*);
2) Ekspertni sustavi (*engl. expert systems*);
3) Neuronske mreže (*engl. neural networks*);
4) Genetski algoritmi (*engl. genetic algorithms*);
5) Evolucijsko programiranje (*engl. evolutionary programming*);
6) Simulirano kaljenje (*engl. simulated annealing*);
7) Tabu algoritam (*engl. tabu search algorithm*);
8) Inteligencija roja čestica (*engl. particle swarm*).

1) Monte Carlo metoda

Bilo koja metoda koja rješava probleme generiranjem velikog broja slučajnih brojeva i promatranjem dijela tih brojeva koji pokazuju željena svojstva naziva se Monte Carlo metoda. Metoda je korisna za rješavanje numeričkih rješenja koji su prekompliciran za rješavanje analitičkim postupcima [4].

2) Ekspertni sustavi

Ovi sustavi se služe iskustvenim (heurističkim) pravilima kojima oponašaju žive eksperte. U principu se pravila utvrđuju temeljem suradnje većeg broja stručnjaka iz promatranog područja, također automatski se mogu određivati i drugim metodama. Ovi sustavi su se pokazali djelotvornim u mnogim praktičnim primjenama [2].

3) Neuronske mreže

Neuronske mreže baziraju se na biološkim neuronskim mrežama, tj. živčanom sustavu. Mreže koje se sastoje od programerski izvedenih neuronskih mreža nazivaju se umjetne neuronske mreže, a od tuda i pojmovi umjetni neuron ili neuronski čvor. Programiranjem se ostvaruje umjetno oponašanje bioloških neurona ili funkcije ljudskog mozga. Pomoću njih se rješavaju problemi linearne optimizacije, problemi nelinearnog programiranja i još mnogi problemi. One mijenjaju rješenje optimizacijskog problema u točku ravnoteže (*engl. equilibrium point*) nelinearnih dinamičkih sustava i mijenjaju optimalni kriterij u energetske funkcije dinamičkih sustava [2].

4) Genetski algoritmi

Genetski algoritmi su prilagodljivi heuristički algoritmi pretraživanja koji su zasnovani na primjeru prirodne selekcije. Temeljni koncept genetskog algoritma je dizajniranje modela procesa utemeljenog na prirodnom sustavu evolucije vodeći se načelom opstanka najjačih. Ovaj algoritam predstavlja inteligentan razvoj slučajnog pretraživanja unutar definiranog prostora kako bi se pronašlo optimalno rješenje problema. Nadmoćan je nad klasičnim metodama i izuzetan za rješavanje problema iz stvarnog svijeta, te je zbog toga dobro

prihvaćen u strojarstvu, računalstvu, elektrotehnici i drugim tehničkim ali ekonomskim i društvenim granama znanosti [5].

5) Evolucijsko programiranje

Evolucijsko programiranje je stohastička metoda optimizacije slična genetskom algoritmu, kojastavlja naglasak na ponašanje povezanosti između roditelja i njihovih potomaka, umjesto da nastoji oponašati specifične genetske operatore kao što je promatrano u prirodi. Evolucijsko programiranje je slično evolucijskim strategijama, iako su dva pristupa razvijena neovisno i samostalno. Ova metoda je koristan način optimizacije kada druge metode kao što su gradijent spuštanja ili izravno analitičko rješenje nisu primjenjivije. Kombinatorno i realno valjane funkcionalne optimizacije, u kojima je optimizacijska površina ili okruženje funkcije cilja hrapava (engl. *fitness landscape*), tj. posjeduju mnoga lokalna optimalna rješenja, dobro se prilagode evolucijskom programiranju. Evolucijsko programiranje ovisi više o mutacijskim i selekcijskim operatorima dok se genetski algoritam uglavnom oslanja na križanja (engl. *crossover*) [1].

6) Simulirano kaljenje

Algoritam simuliranog kaljenja pripada skupini aproksimacijskih algoritama. Algoritam se temelji na stohastičkoj tehnici, ali uključuje mnoga načela algoritma iterativnog poboljšanja. Algoritam je poznat pod nazivom simulirano kaljenje, ali se u literaturi nalazi još i pod nazivima Monte Carlo kaljenje, stohastičko hlađenje, stohastička relaksacija i algoritam slučajne zamjene [2]. Rezultati dobiveni ovim algoritmom su vrlo blizu optimuma, te ne ovise o polaznoj konfiguraciji. Na taj način, metoda simuliranog kaljenja ne pokazuje nedostatke metode iterativnog poboljšanja, te daje bolje rezultate, ali je nešto sporija. U svom originalnom obliku algoritam simuliranog kaljenja se temelji na analogiji između simulacije kaljenja metala i problema kombinatorne optimizacije [2].

7) Tabu algoritam

Tabu pretraživanje je snažan postupak optimizacije koji je uspješno primijenjen na niz kombinatornih problema optimizacije [1]. Tabu pretraživanje posjeduje sposobnost izbjegavanja rješenje u jednom lokalnom minimumu. Tabu algoritam obavlja višestruko

pretraživanje, a zbog sposobnosti pamćenja već obavljenih pretraživanja, ne pretražuje ista područja, što povećava brzinu i pronalazak globalnog optimuma. Najbolja rješenja se dobiju ako se kombinira s drugim metodama.

8) Inteligencija roja čestica

Optimizacija metodom inteligencije roja čestica (*engl. Particle Swarm Optimization)* je stohastička metoda temeljena na populaciji. Metoda je inspirirana društvenim ponašanjem jata ptica ili riba. Te posjeduje sličnosti programiranja s genetskim algoritmom. Ovom metodom sustav se pokreće populacijom slučajnih rješenja i pretraživanja optimuma ažuriranjem generacija. Naspram genetskog algoritma rojenje čestica nema evolucijskih operatora, kao što su križanje i mutacija. U metodologiji optimizacije inteligencijom roja čestica, potencijalna rješenja, zvana čestice, prolaze kroz prostor rješenja prateći trenutne optimalne čestice. Roj čestica kombinira individualni i društveni model uvažavanja najboljih rješenja populacije. Društveni model sugerira da pojedinci zanemaruju vlastito iskustvo i vode se prema uspješnom ponašanju i uvjerenjima pojedinaca u susjedstvu. S druge strane, individualni model tretira pojedinca kao izolirana bića. Čestica pomoću jednog od navedenih modela mijenja svoj položaj u prostoru. Svaka pojedina čestica pamti svoje koordinate, kao i koordinate susjeda kako bi pronašla najbolje rješenje u prostoru problema. Ako čestica uzima cijeli prostor rješenja kao susjedstvo tada je najbolje rješenje ujedno i globalno najbolja vrijednost [1].

3. OPTIMALNA RASPODJELA SNAGE I ANGAŽIRANOST PROIZVODNIH JEDINICA

Problem angažiranosti jedinica (engl. *Unit Commitment Problems*) podrazumijeva određivanje rasporeda i trajanja proizvodnje odabranih proizvodnih jedinice. Angažirane jedinice moraju zadovoljiti opterećenja sustava i zahtjeve rezerve pri minimalnom trošku, ovisno o različitim uvjetima i ograničenjima. Optimalna raspodjela snage (engl. *Ecomomic Dispatch*) bavi se problemom optimalne raspodjele opterećenja među aktivnim jedinicama, zadovoljavajući pritom kriterije ravnoteže snage i ograničenja pogonskih jedinica.

Rješavanje problema angažiranja jedinica pomoću tehnika umjetne inteligencije zahtijeva tri koraka [1]:

- Izraz o problemu ili modeliranje sustava
- Pravila za generiranje probnih rješenja
- Učinkovit algoritam za rješavanje optimalne raspodjele snage

3.1. Pretpostavke mreže

Modeliranje komponenti elektroenergetskog sustava koje utječu na rast sustava je važan korak pri rješavanju problema angažiranosti jedinica. Stupanj detalja tokom modeliranja ovisi o traženoj točnosti i prirodi problema koji se modelira. Osnovne komponente elektroenergetskog sustava uključuju elektrane, transformatore i prijenosnu mrežu. Ovaj dio se odnosi na raspoređivanje upravljivih konvencionalnih izvora zasnivanih na termodinamičkim procesima. Tokom modeliranja pretpostavlja se da je mreža sposobna prenijeti proizvedenu snagu opterećenja bez gubitaka i bez kvarova na mreži. Stoga mreža mora biti savršeno pouzdana. Temeljem toga su napravljene sljedeće inženjerske pretpostavke:

- Razmjena snage između sustava koji se razmatra i sustava koji nisu dio promatrane mreže je fiksna.
- Ne utječe se na opterećenje dodavanjem ili uklanjanjem proizvodnih jedinica.

- Pretpostavlja se da se operativni trošak jedinice za proizvodnju sastoji od tri komponente: troškovi pokretanja, trošak praznog hoda (bez opterećenja) i troškova proizvodnje
- U razmatranju problema angažiranja proizvodnih jedinica, razmatra se koje će rješenje minimizirati ukupni operativni trošak tijekom vremena rada tako da se zadovolje zadana ograničenja [1].

3.2. Pravila za stvaranje probnog rješenja

Za rješavanje kombinacijskih problema optimizacije potrebna su dobra pravila, pomoću kojih se stvara izvedivo pokusno rješenje. Probna rješenja trebaju biti proizvedena nasumce, izvediva i moraju obuhvatiti što veći prostor rješenja problema. Zbog ograničenja problema angažiranih jedinica, to nije jednostavan postupak. Iz dobro postavljenih pravila i dobro odabranog probnog rješenja ubrzava se postupak implementacije pravila, također se i ubrzava generiranje nasumičnih rješenja [1].

3.3. Problemi optimalne raspodjele

Problem optimalne raspodjele snage je bitan problem pri rješavanju problema angažiranja jedinica. Nakon što se generira probno rješenje, odgovarajući trošak rada ovog rješenja izračunava se rješavanjem problema optimalne raspodjele snage. Korištenjem učinkovitog i brzog algoritma za rješavanje optimalne raspodjele poboljšava kvalitetu rješenja angažiranih jedinica, a time i izvedbu ukupnog algoritma za problem angažiranja jedinica. Učinkovit algoritam za rješavanje problema s optimalnom raspodjelom i angažiranjem jedinica naziva se linearno komplementarni algoritam i temelji se na uvjetima Kuhn-Tuckera [1].

3.4. Troškovi proizvodnje

Glavna komponenta operativnih troškova, za temodinamičke i nuklearne jedinice, je trošak proizvodnje snage angažiranih jedinica. Trošak proizvodnje uglavnom je trošak ulaza osnovnog energenta po satu, a održavanje i rad doprinose samo manjoj mjeri. Konvencionalno, trošak proizvodnje jedinice izražava se kao kvadratna funkcija jedinice izlazne snage kako slijedi [1]:

$$F_{it}(P_{it}) = A_i P^2 + B_i P_{it} + C_i \$ / h \quad (3\text{-}1)$$

Gdje su:

$F_{it}(P_{it})$ $F_{it}(P_{it})$- Trošak goriva jedinice *i* u vremenu

(P_{it}) - Radna snaga u MW proizvedena od jedinice *i* u vremenu *t*

$A_i,\ B_i, C_i$ su konstante troškova proizvodnje

3.5. Troškovi pokretanja

Druga komponenta operativnog troška je početni trošak. Trošak pokretanja se definira kao količina energije koja je potrošena da bi se uređaj stavio na mrežu. Trošak pokretanja ovisi o vremenu gašenja jedinice. To se može razlikovati od maksimalne vrijednosti, kada se jedinica pokreće iz hladnog stanja i na mnogo manju vrijednost, gdje je jedinica nedavno isključena.

Izračun troška starta također ovisi o načinu gašenja toplinske jedinice tijekom razdoblja zastoja. Postoje dvije metode za tretman jedinice izvan pogona : način hlađenja i način održavanja. Prva metoda dozvoljava da se kotao jedinice ohladi, a zatim ponovo zagrije do radne temperature kada se ponovno priključuje na mreži. U drugoj metodi, kotao radi te se temperatura održava tijekom isključivanja pomoću dodatne količine energije. Metoda hlađenja se koristi u ovom slučaju zbog njegove praktičnosti jer se primjenjuje u stvarnom vođenju elektroenergetskog sustava. Početni trošak, za jedinicu *i* u vremenu *t*, na temelju metoda hlađenja, formulira se u općenitijem obliku kako slijedi [1] :

$$ST_{it} = So_i\left[1 - D_i \exp(-Toff_i / Tdown_i)\right] + E_i \$ \quad (3\text{-}2)$$

Prema tome, ukupni operativni trošak proizvodnih jedinica u vremenskom periodu (tj. funkcija cilja) angažiranja jedinica) je:

$$F_{it} = \sum_{t=1}^{T} \sum_{i=1}^{N} (U_{it}F_{it}(P_{it}) + V_{it}ST_{it} + W_{it}SH_{it})\$ \quad (3\text{-}3)$$

3.6. Ograničenja

Problem angažiranih jedinica podložan je mnogim ograničenjima ovisno o okolini i parametrima elektroenergetskog sustava u razmatranju. Ograničenja koja se uzimaju u obzir se definiraju kao ograničenja sustava i ograničenja jedinica.

Ograničenja sustava, koja se ponekad nazivaju ograničenjima spajanja, također uključuju kategorije potražnje za opterećenjem i ograničenja rotirajuće rezerve.

Najbitnije ograničenje u angažiranju jedinica je ograničenje potražnje. Navedeno ograničenje znači da generirana snaga svih angažiranih proizvodnih jedinica mora zadovoljiti zahtjeve potražnje tj. opterećenje sustava. Jednadžba aktivne ravnoteže snage je formulirana na sljedeći način [1]:

$$\sum_{i=1}^{N} U_{it}P_{it} = PD_t; _ 1 \leq t \leq T \quad (3\text{-}4)$$

Rotirajuća rezerva (pričuva) je postojanje dostupnih neopterećenih proizvodnih kapaciteta koji su sinkronizirani na sustav te mogu u vrlo kratkom vremenu nadoknaditi manjak snage u mreži [13]. Važno je utvrditi prikladnu raspodjelu rotirajuće pričuve s dva gledišta: zahtjevima pouzdanosti i ekonomskim aspektima. Postoje razne metode za određivanje prijelazne pričuve prema autoru [1].

- Rezerva se izračunava kao postotak predviđenog opterećenja.
- Utvrđeno je da sustav može nadoknaditi gubitak najveće proizvodne jedinice u određenom vremenskom razdoblju.

- Zahtjevi pričuve određeni su kao funkcija pouzdanosti sustava, koja se procjenjuje na temelju vjerojatnosti. Ovdje je rezerva izračunata kao zadani unaprijed određeni iznos koji je postotak očekivanog opterećenja; to je [1]:

$$\sum_{i=1}^{N} U_{it} P_{it} \geq (PD_t + R_t); _ 1 \leq t \leq T \tag{3-5}$$

Ograničenja proizvodnje predstavljaju najmanju snagu ispod kojeg nije ekonomično pokretati pogonsku jedinicu i najveću snagu iznad koje se pogonska jedinica ne smije opteretiti.

$$U_{it} P_{\min i} \leq P_{it} \leq P_{\max i} U_{it} \quad 1 \leq t \leq T, 1 \leq i \leq N \tag{3-6}$$

Minimalno vrijeme isključivanja/pokretanja je također jedno od ograničenja, ako jedinica u pogonu radi, ne može se isključiti prije nego određeno minimalno vrijeme protekne. Ako je jedinica također isključena, ne može se ponovno pokrenuti prije određenog vremena.

Status jedinice (npr. sati uključivanja ili isključivanja) prije prvog radnog sata predloženog rasporedom je važan čimbenik za utvrđivanje da li njegov novi status krši minimalna vremena isključivanja ili paljenja. Također, početni status jedinice utječe na izračun troškova pokretanje [1].

Ako se postrojenje sastoji od dvije ili više generatora, one se ne mogu uključiti istodobno zbog nekih tehničkih uvjeta ili dostupnosti radne snage.

Zbog nekih neuobičajenih stanja, poput prisilnog ispada ili održavanja jedinice, jedinica može postati nedostupna. Pogonska jedinica, agregat, generator, elektrana se također mogu prisiliti na rad kako bi se povećala pouzdanost ili stabilnost sustava, stoga jedinica mora biti u pogonu ili je postavljena na određenu izlaznu snagu. Ograničenje dostupnosti određuje jedinicu ako se nalazi u jednoj od sljedećih situacija: nedostupna, mora biti u pogonu, dostupni ili fiksni izlaz (MW) [1].

Tijekom životnog vijeka jedinice učinak se može mijenjati zbog mnogih uvjeta, na primjer, faktora starenja, okoliša i tako dalje. Ti uvjeti mogu uzrokovati smanjenje proizvodne snage

generatorske jedinice. Posljedično, maksimalne i minimalne granice proizvodne jedinice se mijenjaju [1].

3.7. Pravila za generiranje probnih rješenja

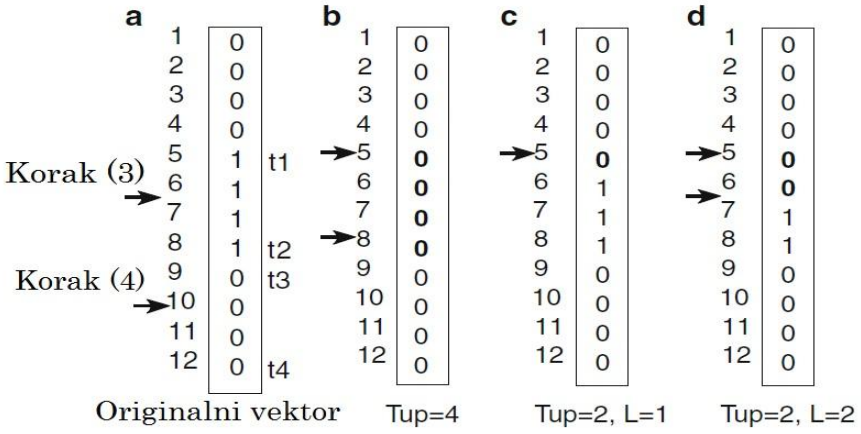
Jedno od najvažnijih problema u rješavanju kombinatornih problema optimizacije jest stvaranje pokusnog rješenja kao susjeda postojećem rješenju. Susjedi trebaju biti generirani slučajnim redoslijedom, izvedivi i rasprostranjeni što je više moguće u cijelom prostoru problema. Tijekom stvaranja izvedivih rješenja, najteže je zadovoljiti ograničenja o najmanjim vremenima isključivanja/pokretanja. Predložena pravila primijenjena su kako bi se probno rješenje smatralo susjedom postojećeg izvedivog rješenja kako je opisano uz pomoć primjera, u sljedećim koracima. Pretpostavljaju se sljedeće vrijednosti: $T = 12$, $Tup_i = 2$ ili 4, i $T_{down} = 1$ ili 4 [1].

1. **korak**: Nasumično generira jedinicu i, $i \sim UD\ (1,\ N)$ i sat t, $t - UD\ (1,\ T)$. Slika 3.1. a) prikazuje status neke jedinice i tijekom razdoblja od 12 h. Jedinica je uključena između razdoblja 5 i 8.

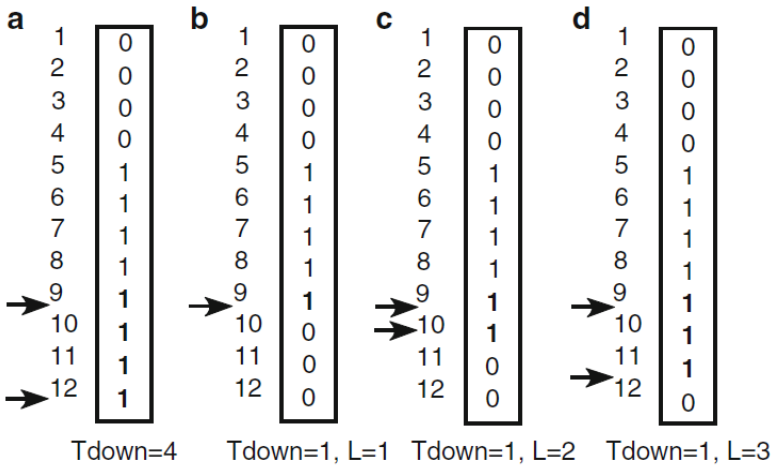
2. **korak**: Ako je jedinica *i* u satu *t* uključena (npr., 5, 6, 7 ili 8 na slici 3.1. a), prelazi se na korak (3) do razmatranja uključenja oko vremena *t*. Inače, ako je jedinica *i* u satu *t* isključena (npr. 1, 2, 3, 4, 9, 10, 11 ili 12), tada se prelazi na korak (4) kako bi se razmotrilo isključenje oko vremena *t*.

3. **korak:** Uključivanje/isključivanje jedinice *i*
   a) Pomaknite se od sata *t* unatrag i naprijed u vremenu, da biste pronašli vrijeme trajanja uključenog perioda. U ovom primjeru ako je $t = 6$, onda je $Ton_i = 8 - 5 + 1 = 4$, a jedinica je uključena tijekom 5,6,7,8 sata.
   b) Ako je $Ton_i = Tup_i$, isključite uređaj u svim satima koji uključuju Ton_i. U primjeru ako je $Tup_i = 4$, tada isključite uređaj u vremenu *t* = 5, 6, 7,8 (slika 3.1b)
   c) Ako je $Ton_i > Tup_i$,tada generiraj $L \sim UD\ (1,\ Ton_i - Tup_i)$.

d) Isključi jedinicu za sate $t1,\ t1\ +\ 1,\ \ldots,t1 + L - 1$, gdje je $t1$ prvi sat u kojem je pogonska jedinica uključena.



Slika 3.1. *Ilustrativni primjer pravila za generiranje probnog rješenja (preuzeto iz [1])*



Slika 3.2. *Ilustrativni primjer pravila za generiranje probnog rješenja (preuzeto iz [1])*

U primjeru ako je $Tup_i = 2$, tada je $L \sim UD(1,\ 2)$. Dakle sljedeća dva moguća rješenja su:

Ako je $L = 1$ tada je jedinica isključena za $t = 5$ (Slika 3.1. c),

ako je $L = 2$, jedinica je ugašena za $t = 5, 6$ (Slika 3.1. d).

4. **korak:** Uključivanje/isključivanje jedinice *i*

a) Pomaknuti se od sata *t* unatrag i naprijed u vremenu, da bi se pronašlo vrijeme isključenja. U primjeru ako je $t = 10$, tada je $T_{off_i} = 12 - 9 + 1 = 4$, onda je pogonska jedinica isključena od 9 do 12 sati.

b) Ako je $T_{off_i} = T_{down_i}$, tada se uključuje jedinica u svim satima T_{off_i}. U primjeru se vidi ako je $T_{down_i} = 4$, tada se pali pogonska jedinica za $t = 9$ do 12 (Slika 3.2. a).

c) Ako je $T_{off_i} > T_{down_i}$, tada generiraj $L \sim UD(1,\ Toff_i - Tdown_i)$

d) Uključiti pogonsku jedincu za $t3, t3 + 1, t3 + 2, \ldots, t3 + L - 1$, gdje je $t3$ prvi sat kada je jedinica isključena.

U primjeru ako je $T_{down_i} = 1$, tada je $L \sim UD(1,\ 3)$. Stoga moguća su tri rješenja:

Ako je $L = 1$, tada je pogonska jedinica uključena za $t = 9$ (Slika 3.2. b),
Ako je $L = 2$, tada je pogonska jedinica uključena za $t = 9,10$ (Slika 3.2. c)
Ako je $L = 3$, tada je pogonska jedinica uključena za $t = 9,10,11$ (Slika 3.2. d).

5. **korak:** provjeriti ograničenja rezerve (pričuve)

Provjeriti da li ograničenja pričuve zadovoljavaju za promijenjena vremenska razdoblja u koracima (3) i (4). Ako je to zadovoljavajuće, tada je moguće probno rješenje; Inače ići do koraka (1) (Slika 3.2).

3.8. Generiranje inicijalnog rješenja

Rješavanje problema angažiranja jedinica pomoću bilo kojeg kombinatornog algoritma za optimizaciju zahtijeva početni izvedivi raspored. Generirano početno rješenje mora biti proizvedeno nasumično i izvedivo. Sljedeći algoritam se koristi za pronalaženje ovog početnog rješenja [1].

Korak (1): Postaviti $U = V = P = 0,\ t = 1$.

Korak (2): Poduzeti sljedeće pod korake:

a) Nasumično generirati jedinicu $i, i \sim UD(1, N)$.

b) Ako je jedinica *i* u satu *t* isključena (Uit = 0), idite na korak (3). Inače se vraća na korak (2a) za odabir druge jedinice.

Korak (3): Slijediti postupak iz koraka (4) u prošlom odlomku, razmotriti uključenje ove jedinice počevši od sata *t*.

Korak (4): Ako je $t = T$, prijeći na korak (5), inače postavite $t = t + 1$ i prijeći na korak (2).

Korak (5): Provjeriti ograničenja pričuve za sva vremena. Ponoviti korake (2) i (3) za sate u kojima ograničenja nisu zadovoljena.

4. METODA ROJA ČESTICA

Ponašanje jata ptica, skupine mrava, jata ribe i dr., bili su polje učenja tijekom povijesti. Takav kolektivni pokret insekata i ptica je poznat kao "ponašanje roja." Biolozi i računalni znanstvenici iz područja umjetne inteligencije počeli su proučavati model biološkog roja za analizu interakcije među društvenim životinjama, kako bi shvatili njihova ponašanja koja bi mogla koristiti za daljnji razvoj tehnologije. Zanimanje inženjera je poraslo, kada se shvatilo da metoda rojenja čestica i umjetne inteligencije brzo i dobro rješava optimizacijske probleme u različitim područjima kao što su telekomunikacijski sustavi, robotika, elektroenergetski sustavi, potrošački aparati, promet i obrasce u transportnim sustavima, vojne aplikacije i mnoge druge. U roju inteligencije, N čestica u roju ili društvenoj skupini se međusobno koordiniraju kako bi postigli određeni cilj svojim kretanjem. Radnje čestica se upravljaju jednostavnim lokalnim pravilima. Inteligentna skupina čestica postiže cilj kroz interakciju cijele skupine. Zbirka akcija proizlazi iz vrste "samoorganizacije" grupe.

Inteligencija roja je skupina kolektivne inteligencije jednostavnih autonomnih agenta. Autonomni agent je podsustav koji komunicira sa svojom okolinom, koja se vjerojatno sastoji od drugih agenata, ali djeluje relativno nezavisno od svih drugih sredstava. Nema globalnog plana ili vođe cijele skupine autonomnih čestica [5].

Primjerice, kretanje ptice u jatu, ptica prilagođava svoje kretnje tako da se koordinira s kretanjem susjeda. Ptica se pokušava kretati zajedno s formacijom jata, tako da njezino kretanje zajedno s drugima u jatu ne uzrokuje raspad formacije te da se izbjegnu sudari među njima. Nema vođe koji će dodijeliti pokrete svakoj ptici, već se ptice pokušavaju koordinirati i kretati same među sobom. Svaka ptica može letjeti na prednjem, srednjem i stražnjem dijelu jata. Oponašanjem roja pomaže pticama da iskoriste nekoliko stvari, uključujući zaštitu od predatora (osobito ptica usred jata) i traženja hrane (u biti svaka ptica iskorištava oči svake druge ptice) [5].

4.1. Opis algoritma

Algoritam roja čestica se koristi za pronalazak globalnog optimuma gdje se rješenja predstavljaju kao točke u *n*-dimenzijskog prostora. Rješenja tj. točke nazivaju se jedinkama. Jedinke su čestice koje su opisane brzinom, položajem u prostoru između svojih susjednih čestica, svaka čestica također sadrži i informaciju o svojoj prošloj najboljoj lokaciji u prostoru. Položaj čestice predstavlja jedno od potencijalnih rješenja koja je dio promatranog n-dimenzionalnog prostora, brzina je vektor u tome jednakom prostoru.

Skupina čestica predstavljena je kao populacija koja svojim kretanjem ovisno o prošlom iskustvu i iskustvu susjednih čestica pretražuje prostor stanja i traži najbolji trenutni položaj. Potrebno je postaviti ograničenja prostora, zbog kretanja čestice kroz prostor postoji mogućnost da se funkcija koja se optimira ne sadrži sve vrijednosti koje čestica poprima kretanjem u prostoru, tj. nije definirana za sve vrijednosti, kako se nebi dogodilo da čestica koja napusti ograničeni prostor ne bude vraćena u njega. Za ograničenja je najbolje postaviti rubne točke prostora, a kako bi se izbjeglo moguće višestruko izlaženje iz prostora treba osigurati i promjenu smjera vektora brzine čestice.

Osim navedenih stavki, važne su i vremenski nepromjenjive stavke kao što su vrijeme eksploracije i eksploatacije kojima algoritam određuje vremena pretraživanja prostora stanja, kao i načine pomoću kojih će se informacije o pronađenim dobrim i kvalitetnim rješenjima prenositi između čestica u prostoru stanja. Prije nego se razmotri točan utjecaj parametara potrebno je formalno opisati algoritam i njegove korake [6].

4.2. Generiranje i inicijalizacija algoritma

Izvođenje algoritma započinje inicijalizacijom čestica, tada se česticama dodjeljuju slučajne vrijednosti položaja i brzine te se definiraju i susjedne čestice tj. susjedstvo. Svaka čestica mora biti iz različitog susjedstva kako bi se ostvarilo dijeljenje informacije o položaju, vrijednosti i vremenu pretraživanja svake pojedine čestice među susjedstvima.

Samo izvođenje algoritma provodi se u diskretnim koracima. Pritom se u svakom koraku za sve čestice računa [6]:

- *novi vektor brzine* zbrajanjem:
  - vektora trenutne brzine,
  - vektora određenog razlikom svojeg najboljeg i trenutnog položaja te
  - vektora određenog razlikom lokalno najboljeg (u susjedstvu najboljeg pronađenog položaja) i trenutnog položaja;
    pomnoženih faktorima povjerenja kako je prikazano izrazom (4-1),
- *novi položaj* dodavanjem novog vektora brzine trenutnom položaju kao u izrazu (4-2) te
- *vrijednost funkcije* za novi položaj; ako je novi položaj bolji od prethodnog najboljeg položaja za tu česticu, vrijednost zadnjeg se zamjenjuje novim položajem.

$$v_i^{k+1} = w \times v_i^{k} + c_1 \times r_1 \times (p_{best_i} - x_i^{k}) + c_2 \times r_2 \times (g_{best} - x_i^{k}) \tag{4-1}$$

$$x_i^{k+1} = x_i^{k} + \chi \times v_i^{k+1} \tag{4-2}$$

Gdje je:

v_i^{k+1}: vektor brzine i^{th} čestice u $(k+1)^{th}$ iteraciji

w : inercija čestice

v_i^{k} : vektor i^{th} čestice k^{th} iteraciji

c_1, c_2 : faktori povjerenja [0, 2.5]

r_1, r_2 : slučajna vrijednost iz uniformne distribucije [0, 1]

p_{best_i} : najbolja pozicija (vrijednost) i^{th} čestice bazirana na iskustvu

g_{best} : Globalno najbolja pozicija čestice u populaciji

x_i^{k+1} : Pozicija i^{th} čestice u $(k+1)^{th}$ iteraciji

x_i^k – *i*-ta komponenta radij-vektora trenutnog položaja čestice,

.

Prikladan odabir mase inercije osigurava dobru ravnotežu između globalnih i lokalnih istraživanja [8].

$$w = w_{\max} - \frac{w_{\max} - w_{\min}}{iter_{\max}} \times iter \tag{4-3}$$

Gdje su:

$w_{\max}$ - vrijednost mase inercije na početku iteracije,

$w_{\min}$ - vrijednost mase inercije na kraju iteracija,

$iter$ - trenutni iteracijski broj i

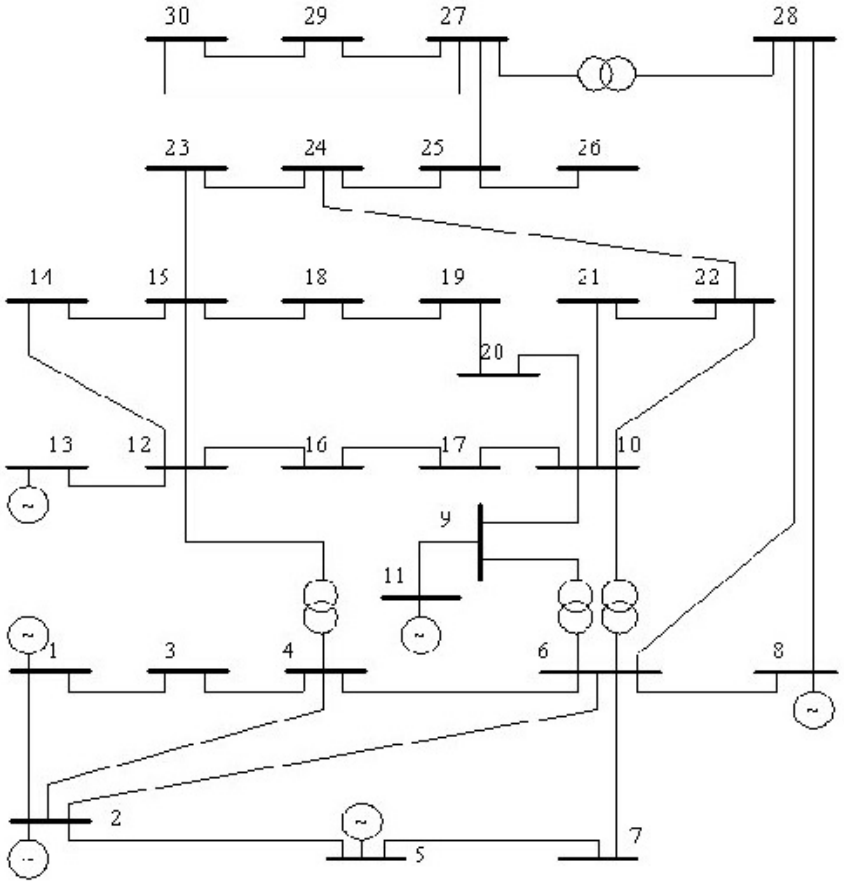
$iter_{\max}$ - maksimalni broj iteracija

5. SIMULACIJE

5.1. Metoda roja čestica (PSO)

Za potrebe simulacije koristio se već izrađen algoritam [7]. Algoritam je napisan u *m* datoteci programskog paketa MATLAB. Za korištenje algoritma bilo je potrebno nadograditi MATLAB datoteke s paketom MATPOWER [9].

Simulacija se pomoću algoritma za PSO optimizaciju „ORPD_30bus“ [7] odrađuje za slučaj mreže testni IEEE elektroenergetski sustav od 30 čvorova, također se datoteke „runpf“ i „printpf“ moraju prilagoditi PSO simulaciji u paketu Matpower kako je autor algoritma Kartik S. Pandya zadao. Zadanim slučajem u paketu *MATPOWER 3.2* [9] odrađujemo simulaciju tokova snaga za 30 sabirnica i 6 generatora, 21-o opterećenje, 41-u granu i 4 transformatora. Na slici 5.1. možemo vidjeti jednopolnu shemu mreže „case_ieee30“.



Slika 5.1. *Jednopolna shema "case_ieee30" slučaja mreže (preuzeto iz [10])*

Generatori su kontrolne varijable *vg1, vg2, vg5, vg8* i *vg13*, položaji transformatorskih preklopki su kontrolne varijable *T1, T2, T3, T4*, a injektirana jalova energija kompenzacijskih kondenzatora su kontrolne varijable *QC3, QC10* i *QC24*. Kontrolne varijable su početni podaci određenih dijelova mreže kojima se u n-dimenzionalnom prostoru traži najbolji položaj kako bi se dobilo najbolje tj. optimalno rješenje u našem slučaju smanjenje radne i jalove (reaktivne) snage u mreži.

Obavljene su simulacije za 10, 20, 50, 100 i 200 iteracija. Procedura obavljanja algoritma je objašnjena u sljedećim koracima gdje je vidljivo da se za simulaciju koristi i Newton-rahpsonovu metodu optimizacije koja se koristi prilikom optimizacije tokova snaga s ograničenjima koja se mogu prevesti u optimizacijske probleme bez ograničenja kako bi se ubrzao i pojednostavio algoritam. Proračun tokova snaga obavljen je Newton-Raphson iterativnim postupkom koji je zadan datotekom „runpf“ [9].

Procedura izvođenja algoritma po koracima [7]:

1. Definiranje kontrolnih varijabli (*vg1, vg2, vg5, vg8, vg11, vg13, T1, T2, T3, T4, QC3, QC10* i *QC24*), unesti broj iteracija za PSO algoritam
2. Postaviti varijablu *iter* u nulu
3. Nasumično generirati generaciju čestica i njihove brzine
4. Za svaku česticu pokrenuti Newton-Raphson algoritam tokova snaga kako bi se saznali gubici
5. Izračun funkcije dobrote svake čestice
6. Pronalazak lokalnog i globalnog najboljeg rješenja *(Pbest* i *Gbest)* pomoću funkcije dobrote
7. Svakom iteracijom varijabla *iter* se povećava za 1 ($iter + 1$)
8. Računanje vektora brzine svake čestice, prilagodba zadanim uvjetima
9. Računanje nove pozicije čestice
10. Za novu česticu pokrenuti Newton-Raphson algoritam tokova snaga kako bi se saznali gubici
11. Izračun funkcije dobrote svake nove čestice
12. Usporedba kvalitete svake čestice (*P*) sa starim *Pbest*, ako je *P* bolji tada je *Pbest=P*
13. Postaviti *Pbest* za *Gbest*
14. Ponavljati do sedmog koraka sve dok sve iteracije nisu završene

15. Koordinate Gbest čestice daju optimiziranu vrijednost kontrolne varijable, te njegova funkcija procjene kvalitete daje minimalnu vrijednost gubitaka

Prije izvođenja simulacije potrebno je postaviti parametre metode inteligencije roja čestica, u simulaciji su korišteni parametri koje je zadao autor algoritma. Parametri su mogu mijenjati u samom programu po želji. Korišteni parametri i njihove vrijednosti su navedeni u nastavku:

Veličina populacije = 50

Konstanta ubrzanja (C1 i C2) = 2.1 i 2.0

Konstanta ubrzanja (C1 i C2) = 2.1 i 2.0

C1 – individualna konstanta ubrzanja ; C2 - društvena konstanta ubrzanja

Faktor ograničenja brzine čestice = 0.729

Maksimalna i minimalna vrijednost inercije čestice = 1 i 0.2

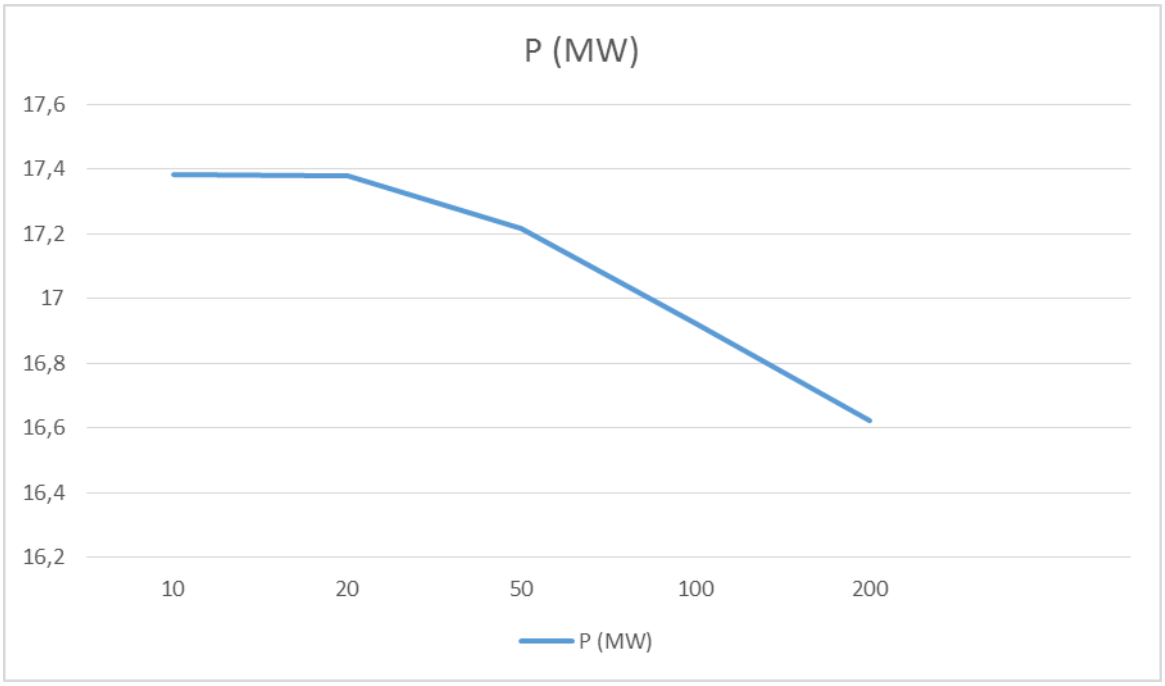
Maksimalna i minimalna brzina čestice = 0.003 i -0.003

Nakon obavljanja svih iteracija zadanih u simulaciju dobiju se rezultati u datoteci „diary“. Iz navedene datoteke iščitavaju se dobiveni podaci za gubitke radne i jalove snage. Nakon obavljene simulacije za 200 iteracija možemo primijetiti da je već nakon 100 iteracija promjena rezultata neznatna. Što je vidljivo u tablici 5.1, i na slikama 5.2. i 5.3.

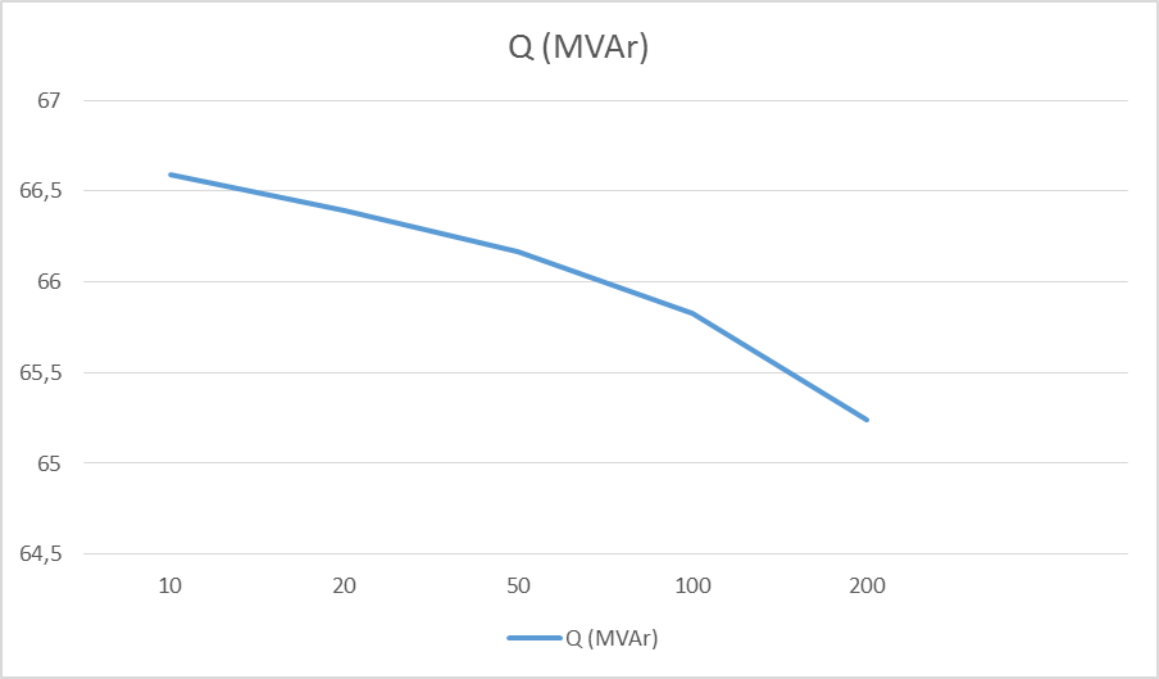
U izvođenju simulacija za isti broj iteracija svaki put se dobije drugačije rješenje, koje je blizu ostalih rezultata ali ne identično, vrlo rijetko se dobije jako različito rješenje, to se događa zbog toga što je metoda inteligencije roja čestica stohastička. Rezultati simulacije se mogu vidjeti u tablici 5.1.

Tablica 5.1. *Rezultati simulacije*

Broj iteracija	P	Q
10	17,3812	66,59
20	17,3785	66,39
50	17,216	66,17
100	16,9234	65,83
200	16,6231	65,22



Slika 5.2. *Grafički prikaz smanjenja gubitaka radne snage za povećanje broja iteracija*



Slika 5.3. *Grafički prikaz smanjenja gubitaka jalove snage za povećanje broja iteracija*

Tablica 5.2. *Usporedba rezultata simulacije*

		Gubici prije optimizacije	Gubici poslije optimizacije	Smanjenje gubitaka u postocima (%)
1.	Gubici radne snage (MW)	17.53	16.62	6.2
2.	Gubici jalove snage (MVAR)	67.59	65.22	3.5

U tablici 5.2. vidljivo je smanjenje gubitaka nakon optimizacije. Metodom roja čestica brzo se dolazi do rezultata koji su zadovoljavajući, smanjenje gubitaka radne snage je za 0,91 MW, te smanjenje jalove snage je 2,37 MVAR. Simulacija je od pomoći pri optimizaciji mreže, tako da se brzom simulacijom daje uvid u mogućnost smanjenja gubitaka.

U datoteci „diary“ koja se generira izvođenjem algoritma možemo točno zaključiti na kojoj sabirnici, grani ili generatoru možemo napraviti promjenu kako bi smanjili gubitke.

Zaključuje se da najveći gubici postoje na grani između sabirnica 1 i 2, najmanji napon je na sabirnici 30 kao i najmanji kut.

Nakon što se mreža upozna i kada se obavi dovoljan broj iteracija, vrlo lako se zaključuje gdje su nedostaci mreže. Stoga se ovim radom žele pojasniti metode optimiranja mreže, ponajviše metoda roja čestica. U radu su detaljno prikazani koraci koji se moraju obaviti kako bi se pomoću programskog paketa *MATLAB* mogla izvršiti optimizacija metodom inteligencije roja čestica. U radu je za primjer mreže korišten slučaj „case_ieee30“ koji je standardan pri istraživanjima, ali algoritam „ORPD_30bus“ se ne može koristiti i za druge mreže koje su zadane preko m-datoteku u matlabu, osim ako se ne izvrše veće promjene u algoritmu i da se odrede kontrolne varijable.

5.2. Optimizacija pomoću Newton-Raphson i Gauss- Seidel metode

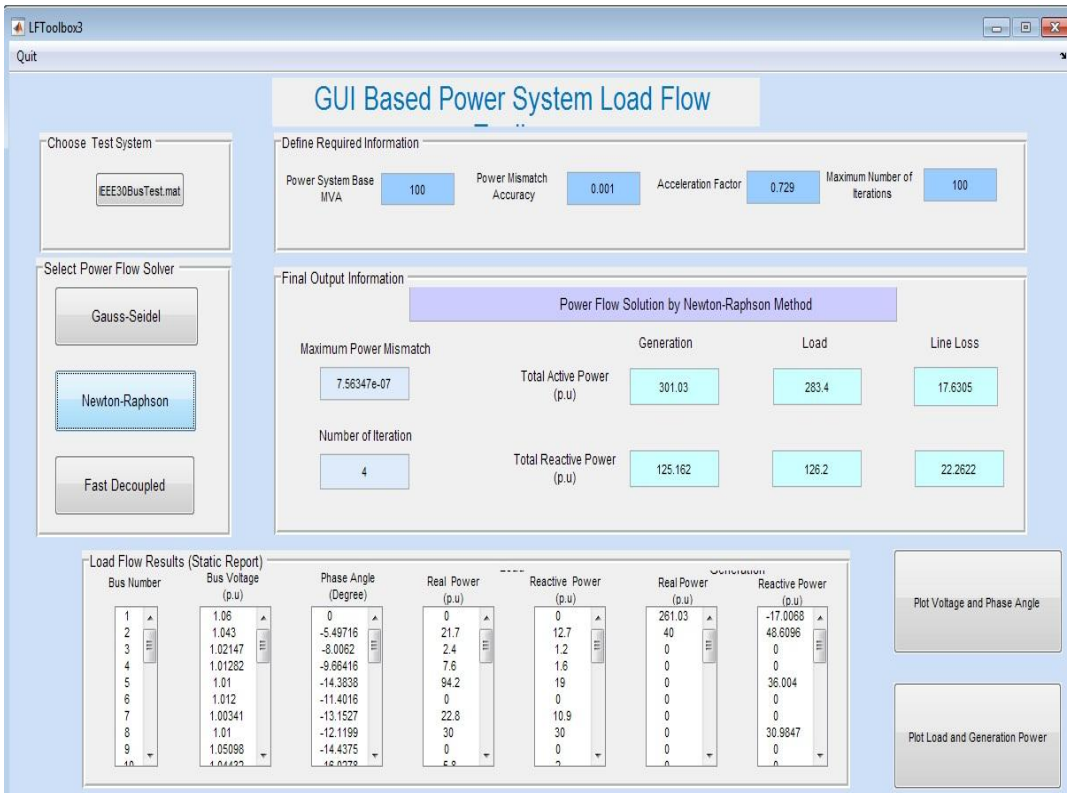
Za sljedeću simulaciju je korištena aplikacija za MATLAB autora KyawMyoLin, profesora na sveučilištu *MandalayTechnological University*-a [11].

Aplikacija vrši simulacije za optimiranje tokova snaga kao i prethodno navedeni algoritam, također se koristi Newton-Raphson metoda u jednom slučaju, a u drugome Gauss-Seidel metoda.

Za ovu simulaciju se također koristi primjer mreže „case_ieee30“.

Prvotno se mora u grafičkom sučelju aplikacije zadati početna mreže, bazna snaga mreže, točnost i broj željenih iteracija.

Ako zadani broj iteracija nije dovoljan javit će se greška da funkcija nije konvergirala za zadani broj iteracija.



Slika 5.4. *Početni zaslon grafičkog sučelja*

Kada se unesu svi podaci potrebno je odabrati metodu optimiranja. Prvotno se odabire Newton-Rahpson-ova metoda.

Nakon simulacije u workspace-u iščitava se tablica koja sadrži broj sabirnice u mreži, napon na sabirnici u p.u. , kut napona, proizvodnju i potrošnju.

5.2.1. Rezultati za Newton-Raphson metodu

Tablica 5.3. *Rezultati simulacije*

Bus	Voltage	Angle	------Load------		---Generation---		Injected
No.	Mag.	Degree	MW	Mvar	MW	Mvar	Mvar
1	1.060	0.000	0.000	0.000	261.030	-17.007	0.000
2	1.043	-5.497	21.700	12.700	40.000	48.610	0.000
3	1.021	-8.006	2.400	1.200	0.000	0.000	0.000
4	1.013	-9.664	7.600	1.600	0.000	0.000	0.000
5	1.010	-14.384	94.200	19.000	0.000	36.004	0.000
6	1.012	-11.402	0.000	0.000	0.000	0.000	0.000
7	1.003	-13.153	22.800	10.900	0.000	0.000	0.000
8	1.010	-12.120	30.000	30.000	0.000	30.985	0.000
9	1.051	-14.437	0.000	0.000	0.000	0.000	0.000
10	1.044	-16.028	5.800	2.000	0.000	0.000	19.000
11	1.082	-14.437	0.000	0.000	0.000	16.137	0.000
12	1.057	-15.306	11.200	7.500	0.000	0.000	0.000
13	1.071	-15.306	0.000	0.000	0.000	10.433	0.000
14	1.042	-16.195	6.200	1.600	0.000	0.000	0.000
15	1.038	-16.282	8.200	2.500	0.000	0.000	0.000
16	1.045	-15.883	3.500	1.800	0.000	0.000	0.000
17	1.039	-16.192	9.000	5.800	0.000	0.000	0.000
18	1.028	-16.887	3.200	0.900	0.000	0.000	0.000
19	1.025	-17.056	9.500	3.400	0.000	0.000	0.000
20	1.029	-16.856	2.200	0.700	0.000	0.000	0.000
21	1.032	-16.472	17.500	11.200	0.000	0.000	0.000
22	1.033	-16.458	0.000	0.000	0.000	0.000	0.000
23	1.027	16.666	3.200	1.600	0.000	0.000	0.000
24	1.022	-16.834	8.700	6.700	0.000	0.000	4.300
25	1.019	-16.428	0.000	0.000	0.000	0.000	0.000
26	1.001	-16.846	3.500	2.300	0.000	0.000	0.000
27	1.026	-15.916	0.000	0.000	0.000	0.000	0.000
28	1.011	-12.061	0.000	0.000	0.000	0.000	0.000
29	1.006	-17.140	2.400	0.900	0.000	0.000	0.000
30	0.994	-18.019	10.600	1.900	0.000	0.000	0.000
Total			283.400	126.200	301.030	125.162	23.300

Iz podataka sabirnica se zaključuje da je najveća proizvodnja radne snage na sabirnici 1, najveće opterećenje je na sabirnici 5.

Nakon pregleda podataka na sabirnicama pregledavaju se gubici između čvorova tj. sabirnica. Zbog previše podataka iščitati će se potrebni podaci koji se nalaze u tablici 5.4. i u tablici 5.5.

Tablica 5.4. *najveći gubici u mreži nakon optimiranja*

Između čvorova	**Gubitak radne snage (MW)**	**Gubitak jalove snage (Mvar)**
1 i 2	5.464	10.525
1 i 3	2.810	7.092
2 i 5	2.997	8.184

Tablica 5.5. *Preklopke transformatora u mreži nakon optimizacije*

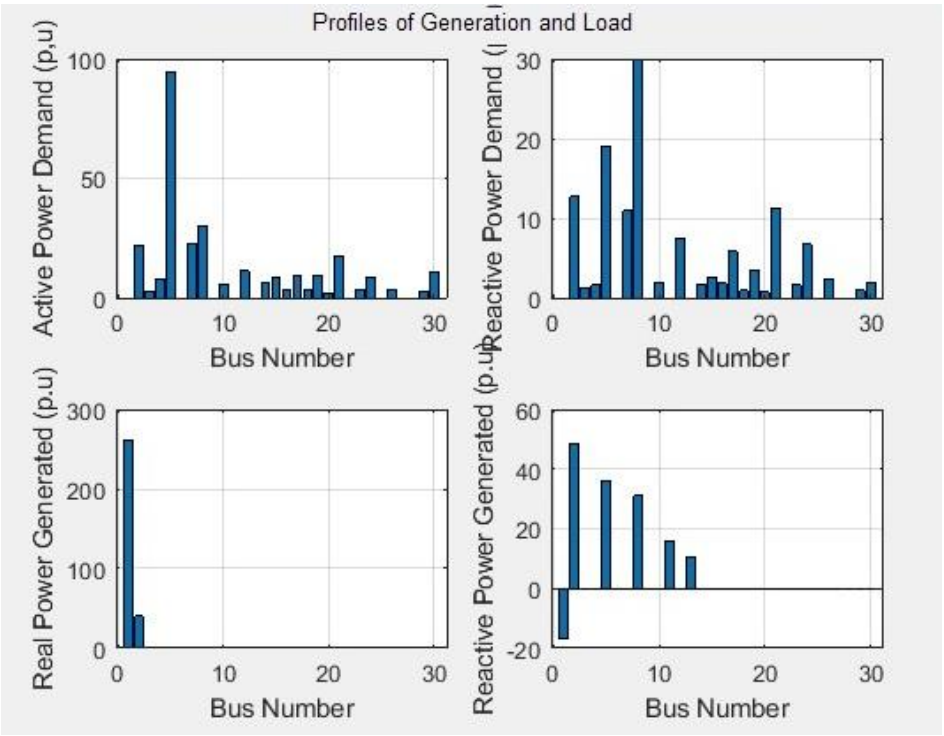
Položaj transformatora	**Položaj preklopke transformatora (p.u.)**	**Gubici jalove snage na vodu (MVar)**
$T_{4\text{-}12}$	0.932	4.685
$T_{6\text{-}10}$	*0.978*	*1.594*
$T_{6\text{-}28}$	*0.969*	*1.278*
$T_{28\text{-}27}$	*0.968*	*1.309*

Tablica 5.6. *Usporedba rezultata simulacije*

1.	Gubici radne snage (MW)	17.55 MW
2.	Gubici jalove snage (MVAR)	22.42 MVar

Zaključuje se da su rezultati Newton-Rahpsonovom metodom lošiji nego metodom inteligencije roja čestica iz prvog primjera.

Na slici 5.5. može se vidjeti snage na svim sabirnicama u mreži.



Slika 5.5. *Grafički prikaz potrošnje i generiranja radne i aktivne snage po sabirnicama*

5.2.2. Rezultati za Gauss- Seidel metodu

Rezultati za Gauss-Seidel metodu se dobivaju na isti način kao i za prethodnu metodu, jer su dio istog alatnog paketa.

Simulaciju pomoću ove metode smo izvršili pomoću istih parametara kao i za prethodne dvije simulacije.

Dobiveni rezultati se nalaze u sljedećim tablicama.

Tablica 5.7. *najveći gubici u mreži nakon optimiranja*

Između čvorova	**Gubitak radne snage (MW)**	**Gubitak jalove snage (Mvar)**
1 i 2	5.090	9.406
1 i 3	2.599	6.228
2 i 5	2.913	7.831

Tablica 5.8. *Preklopke transformatora u mreži nakon optimizacije*

Položaj transformatora	**Položaj preklopke transformatora (p.u.)**	**Gubici jalove snage na vodu (MVar)**
$T_{4\text{-}12}$	0.932	4.346
$T_{6\text{-}10}$	*0.978*	*1.388*
$T_{6\text{-}28}$	*0.969*	*1.124*
$T_{28\text{-}27}$	*0.968*	*1.238*

Primjećuje se da nakon optimizacije nema promjena na preklopkama transformatora od rezultata u prethodnoj simulaciji, ali gubici jalove su manji nego za prošlu simulaciju.

Tablica 5.8. *Usporedba rezultata simulacije*

1.	Gubici radne snage (MW)	16.618 MW
2.	Gubici jalove snage (MVAR)	18.08 MVar

Primjećujemo da su gubici radne i jalove snage manji u odnosu na prethodnu metodu i to za 10%, razlog tomu je drugačije određivanje funkcije cilja kao i različiti načini konvergencije rješenja.

Stoga se preporučuje upotreba prvog algoritma za izvođenje simulacije, tj. algoritam autora Kartik-a S. Pandya , zbog toga što je provedbom simulacije pomoću tog algoritma dobivena srednja vrijednost gubitaka. Točnost svake metode nije potpuno povjerljiva stoga su za ozbiljnije potrebe potrebna daljnja razmatranja, također se preporuča provjera rezultata na jednom od komercijalno dostupnih alata kao što je PowerWorld programski paket [15].

6. ZAKLJUČAK

Diplomski rad za cilj je imao obraditi metode optimizacije tokova snaga elektroenergetske mreže, odabrati jednu od metoda i izraditi simulaciju baziranu na toj metodi. Odabrana metoda je „Metoda roja čestica“ ili kraće PSO (*engl. Particle Swarm Optimization*). Metoda je odabrana zbog sve češće primjene u optimizaciji elektroenergetskih sustava, zbog vrlo visoke preciznosti podataka te ponajviše zbog kratkog vremena rješavanja problema.

Za simulaciju se koristio programski paket MATLAB, u kojem je korišten MATPOWER programski paket. MATPOWER je paket m-datoteka koje služe za rješavanje problema tokova snaga. Prije početka simulacije moraju se unesti vrijednosti parametara simulacije radi dobivanja što preciznijeg i bržeg rješenja. Također, potrebne su i dodatne promjene u datotekama MATPOWER-a kako bi simulacija ispravno radila. Simulacija koristi Newton-Raphsonovu metodu za proračun tokova snaga nakon svake iteracije zadane metodom inteligencije roja čestica.

Kako bi se potvrdili rezultati prvotne simulacije izvedene su još dvije dodatne simulacije. Korištena je aplikacija za MATLAB autora KyawMyoLin. Prva simulacija u poglavlju 5.2. ima Newton-Raphson metodu konvergencije kao i prvotna metoda, a druga koristi Gauss- Seidel metodu.

Konačni rezultati svih triju simulacija imaju odstupanja do maksimalno 10%. Prvotna simulacija smatra se najispravnijom zbog toga što su rezultati te simulacije po iznosima između rezultata druge i treće simulacije. Također se preporuča provjera rezultata na jednome od komercijalno dostupnih alata kao što je PowerWorld programski paket.

Simulacije upućuju na čvorove na kojima je moguće napraviti određene promjene kako bi se gubici smanjili, a time povećala iskoristivost mreže.

7. LITERATURA

[1] Soliman Abdel-Hady Soliman, Abdel-Aal Hassan Mantawy, Modern Optimization Techniques with Applications in Electric Power Systems, Springer Science+Business Media, New York, 2012.

[2] Amir Softić, Poboljšani model za planiranje niskonaponskih distribucijskih mreža kombiniranjem hijerarhijskih i stohastičkih optimizacijskih metoda, Sveučilište Josipa Jurja Strossmayera u Osijeku, Elektrotehnički fakultet Osijek, doktorska disertacija, Osijek, 2016.

[3] Frano Tomašević , Analiza i usporedba metoda optimizacije tokova jalove snage u elektroenergetskom sustavu, Sveučilište u Zagrebu, Fakultet elektrotehnike i računarstva Zavod za visoki napon i energetiku
URL:
https://www.researchgate.net/publication/265098057_Analiza_i_usporedba_metoda_optimizacije_tokova_jalove_snage_u_elektroenergetskom_sustavu
(Datum pristupa: 22.07.2017.)

[4] Weisstein, Eric W. "Monte Carlo Method." From MathWorld--A Wolfram Web Resource URL: http://mathworld.wolfram.com/MonteCarloIntegration.html
(Datum pristupa: 27.07.2017.)

[5] S. Sumathi, Surekha Paneerselvam, Computational intelligence paradigms : theory & applications using MATLAB, CRC Press ,Taylor and Francis Group, Boca Raton, 2010.

[6] Denis Ćutić, Programsko ostvarenje algoritama rojeva čestica, Sveučilište u Zagrebu, Fakultet elektrotehnike i računarstva, diplomski rad, Zagreb, 2014

[7] Kartik S. Pandya, Optimal Reactive Power Dispatch (ORPD), Using Particle Swarm Optimization (PSO), Charotar University of Science and Technology, 2013.

[8] B. Zhao, C.X. Guo, Y.J. Cao, A Multiagent-Based Particle Swarm Optimization Approach for Optimal Reactive Power Dispatch, IEEE Trans. On Power Systems, 2005.

[9] MATPOWER paket

URL: http://www.pserc.cornell.edu/matpower/ (Datum pristupa: 26.6.2017.)

[10] Prikaz elektroenergetskog sustava sa 30 sabirnica

URL: http://al-roomi.org/multimedia/Power_Flow/30BusSystem/IEEE30BusSystemB.jpg

(Datum pristupa 12.11.2017.)

[11] Paket alata za MatLab, GUI based load flow toolbox, Kyaw Lin, 2018.

URL: https://www.mathworks.com/matlabcentral/fileexchange/64367-gui-based-load-flow-toolbox (Datum pristupa 23.2.2018.)

[12] Dinko Mandarić, Optimiranje upravljanja preklopke distributivnog transformatora i snage distribuiranih izvora evolucijskim algoritmom, Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek, diplomski rad, Osijek, 2016.

URL: https://urn.nsk.hr/urn:nbn:hr:200:135266 (Datum pristupa 28.11.2017.)

[13] Doc. dr.sc.Igor Kuzle FER- Zagreb, Planiranje pomoćnih usluga, 8. simpozij o sustavu vođenja EES-a, Cavtat, 2008.

URL: https://bib.irb.hr/datoteka/369067.Pozvani_referat.pdf (Datum pristupa 28.11.2017.)

[14] PowerWorld, programski paket

URL: https://www.powerworld.com/ (Datum pristupa 16.03.2018.)

[15] Ivan Šimić, Sistematizacija metoda nelinearnog programiranja, diplomski rad, Sveučilište u Zagrebu, Fakultet strojarstva i brodogradnje, Zagreb, 2010.

SAŽETAK

Povećanje svakodnevnih potreba ljudi za električnom energijom raste ubrzano, ponajviše zbog sve veće svijesti o ekologiji i klimatskim promjenama kao i cijenama naftnih derivata. Posljedica toga su i mnogobrojni raznovrsni distribuirani izvori energije koji mijenjaju tokove snaga na mreži. Zbog raspršene i kompliciranе proizvodnje potrebno je elektroenergetski sustav što bolje optimirati kako bi se što bolje iskoristila infrastruktura i smanjili troškovi. Pomoć pri optimiranju pružaju razni algoritmi i metode optimizacije mreže i tokova snaga koji su opisani u radu. U ovome radu koristila se metoda roja čestica za optimiranje zadane mreže u programskom paketu Matlab pomoću nadogradnje Matpower. Dobiveni rezultati pokazuju iznos smanjenja gubitaka i upućuju na mjesta gdje postoji mogućnost smanjenja gubitaka radne i jalove snage. Rezultati simulacije su okvirno pomagalo gdje smanjiti gubitke, a za realnu upotrebu bilo bi potrebno dodatno istražiti rješenja jer su prilikom simulacije poneka realna ograničenja zanemarena radi jednostavnosti.

Ključne riječi: električna energija, optimiranje, distribuirani izvori, tokovi snaga, elektroenergetski sustav, PSO, metoda roja čestica,

Application of optimization methods in the electric power system

SUMMARY

Increasing the daily needs of people for electricity is growing rapidly, mainly due to the increasing awareness of ecology and climate changes, as well as the prices of petroleum products. As a result, there are many different distributed energy sources that change power flows on the network. Due to scattered and complicated production, the power system needs to be optimized, to reduce the infrastructure costs. Optimization help to provide a variety of algorithms and network optimization methods and power flows that are described in the work. In this paper the particle swarm method was used to optimize the default network in the Matlab software package by using Matpower upgrade. The obtained results show the amount of loss reduction and point to places where there is a possibility of reducing true power and reactive power losses. The simulation results are a cost-effective way to reduce the losses, for realistic usefurther research would be necessary, because some realistic constraints are ignored during simulation for simplicity.

Key words: electrical energy, optimization, distributed sources, power flow, electric power system, PSO, particle swarm method,

ŽIVOTOPIS

HRVOJE PALČIĆ

Rođen je 4. Srpnja 1993. u Virovitici. Osnovnu školu pohađa u Suhopolju. Završava osnovnu školu 2008., iste godine upisuje srednju tehničku školu u Virovitici, smjer elektrotehničar. Tokom školovanja sudjeluje na županijskim natjecanjima iz matematike, te ostvaruje izvrstan uspjeh tokom sve 4 godine. Sudjelovao je na državnoj smotri završnih radova 2012. godine, gdje je osvojio drugo mjesto.

2012. godine upisuje preddiplomski sveučilišni studij elektrotehnike na Elektrotehničkom fakultetu u Osijeku. Tečno govori i piše engleski jezik. Informatički je pismen te se izvrsno služi programskim paketom Microsoft office (Excell, Word, PowerPoint, Visio) i AutoCAD-om, programskim paketom MATLAB, Easy Power i drugi. Tokom studiranja povremeno radi za GATO-automatiku, gdje svoja znanja primjenjuje u praksi i usavršava svoje znanje.

Nakon završetka preddiplomskog studija 2015. godine, iste godine upisuje i diplomski studij elektrotehnike na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija u Osijeku, namjera mu je po završetku studija raditi u struci.