



**UNIVERZITET U NOVOM SADU
FAKULTET TEHNIČKIH NAUKA
U NOVOM SADU**



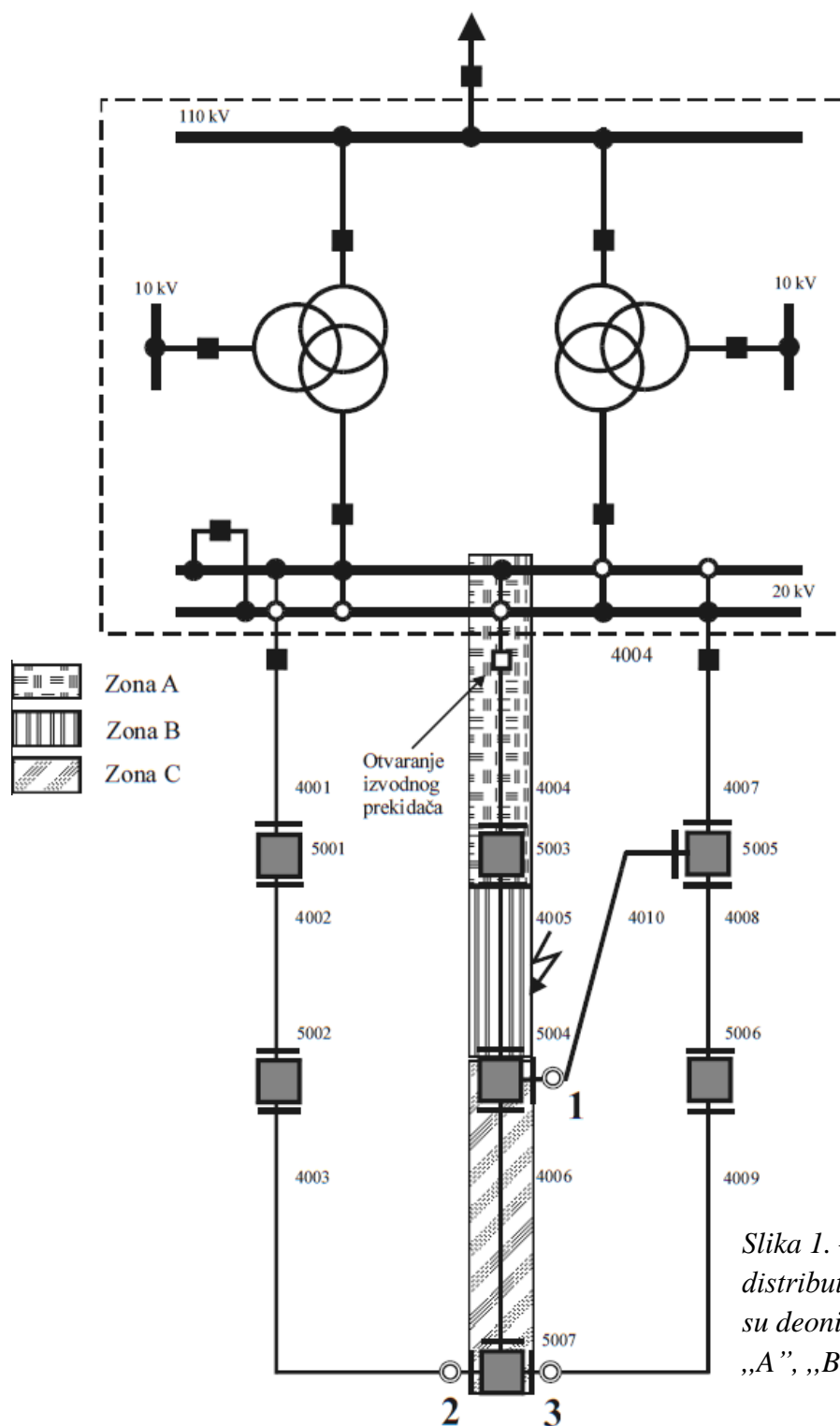
ALGORITAM ZA IZOLOVANJE KVARA I RESTAURACIJU NAPAJANJA

SADRŽAJ

Zadatak	3
Restauracija napajanja	4
Metodologija za restauraciju napajanja	5
Višekriterijumski algoritam	7
1. Određivanje zone „C”	7
2. Određivanje lokalne mreže	8
3. Selekcija varijanti rezervnog napajanja	9
Opis programa	9
<i>Module_Elements.f90</i>	10
<i>Module_Nodes.f90</i>	10
<i>Module_Branches.f90</i>	10
<i>Module_Load_Grid.f90</i>	11
<i>Module_Power_Restoration.f90</i>	11
<i>Module_Recursion.f90</i>	12
<i>Module_Printer.f90</i>	13
<i>Main.f90</i>	14

ZADATAK

Razmatra se deo jednostavnog elektroenergetskog sistema na slici 1.
U datoj distributivnoj mreži simuliran je kratak spoj na deonici koja je na slici 1. označena sa 4005.



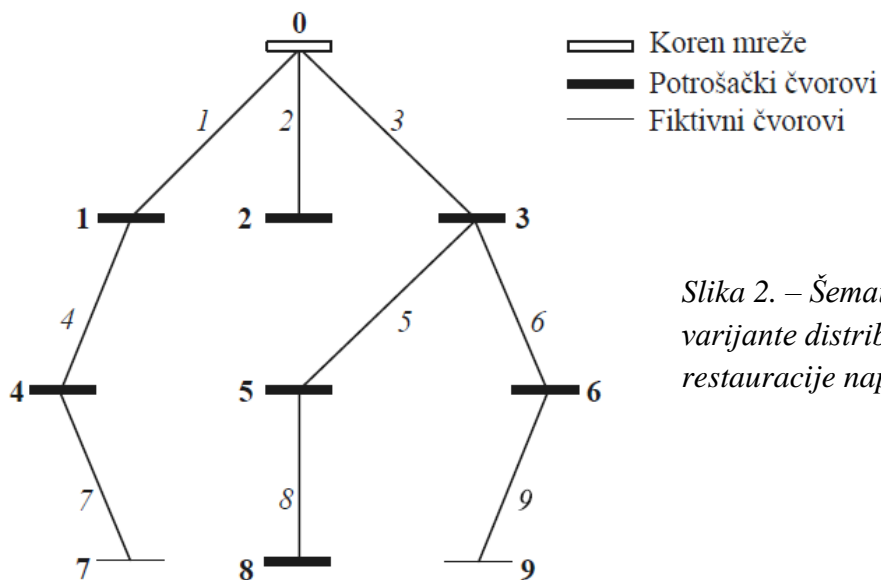
Slika 1. – Razmatrana distributivna mreža. Prikazane su deonice sa kvarom, zone „A”, „B” i „C”.

Svi neophodni podaci o sistemu nalaze se u spoljašnjoj, ulaznoj datoteci.

Napisati program u programskom jeziku Fortran koji:

1. Učitava podatke o sistemu iz ulaznih datoteka
2. Zaustavlja rad prikazane mreže i izoluje kvar
3. Deli datu mrežu na zone „A”, „B” i „C”
4. Vrš restauraciju napajanja za zonu „A” i „C”
5. Prikazuje varijacije izgleda mreže nakon restauracije napajanja u izlaznoj datoteci

Izgled rešenja, odnosno šematski prikaz distributivne mreže u jednoj od tri varijante nakon restauracije napajanje prikazan je na slici 2. Na ovoj šemi, čvorovi i grane nisu obeležene odgovarajućim indeksima kao u postavci zadatka, što može da se zanemari. Takođe, prikazani su fiktivni čvorovi što u rešenju zadatka nije neophodno prikazati.



Slika 2. – Šematski prikaz jedne varijante distributivne mreže nakon restauracije napajanja

RESTAURACIJA NAPAJanJA

Restauracija napajanja je funkcija koja se koristi za određivanje optimalnog plana manipulacija za obnavljanje napajanja na delu izvoda koji je ostao bez napajanja nakon izolacije kvara. Funkcija kao rešenje daje predlog svih varijanti alternativnog napajanja. Varijante alternativnog napajanja se rangiraju u skladu sa korisnički specificiranim kriterijumom. Izbor kriterijuma zavisi od osobina razmatrane distributivne mreže i tipa problema koji se rešava restauracijom.

Prvo će biti data opšta postavka problema restauracije napajanja distributivne mreže i metodologije za njeno rešavanje. Zatim će biti prikazan jedan specijalizovan algoritam za rešavanje problema restauracije napajanja, kojim će gore postavljeni zadatak biti rešen.

METODOLOGIJA ZA RESTAURACIJU NAPAJanJA

Problem određivanja plana restauracije distributivne mreže je u osnovi višekriterijumski kombinatorni nelinearni problem sa ograničenjima. Pored toga, u praktičnoj primeni ovaj problem je izuzetno velike dimenzionalnosti, što ga dodatno usložnjava. Za rešavanje problema restauracije koristi se pet grupa metoda:

1. Optimizacione metode
2. Heurističke metode
3. Ekspertni sistemi i veštačka inteligencija
4. Probablističke metode i metode bazirane na „fuzzy” pristupu
5. Metode bazirane na tehnikama upravljanja sa rizikom

Optimizacione metode se koriste za rešavanje različitih pogonskih problema u cilju određivanja optimalnog plana restauracije distributivne mreže sa aspekta zadatog optimizacionog kriterijuma. Uobičajeno se koriste: mešovito celobrojno programiranje, mrežno programiranje, metoda grananja i ograničavanja, itd. Dobre osobine ovih optimizacionih metoda su što se problem može precizno definisati u smislu cilja (kriterijumske funkcije) i skupa ograničenja koje je potrebno ispoštovati. Na primer, kriterijumska funkcija se najčešće definiše kao neki od sledećih ciljeva: minimalan iznos snage nerestauriranog opterećenja, minimalni troškovi manipulacija, najbolji balans opterećenja po napojnim transformatorima i izvodima, itd. Nedostatak ovih metoda je što je priroda problema toliko složena, da se svi aspekti ne mogu precizno uzeti u obzir. Na primer, određena optimalna rešenja su praktično teško izvodljiva u praksi. Zatim, često je efikasnije realizovati jednostavniju varijantu rezervnog napajanja sa malim brojem manipulacija i neznatnom redukcijom opterećenja, nego izvršiti složene manipulacije po dubini mreže kojima se obezbeđuje napajanje svim potrošačima. Dalje, velika dimenzionalnost problema zahteva dugotrajnu računarsku obradu u realnom vremenu, čak i sa najsavremenijim računarima. Konačno, problemi konvergencije su takvi da se često kao rezultat dobijaju samo suboptimalna rešenja.

Heurističke metode su bazirane na originalnim algoritmima za određivanje plana restauracije distributivnih mreža. Ovi algoritmi su razvijeni na bazi maksimalnog poznavanja karakteristika distributivne mreže (radijalnost, višestruke alternativne mogućnosti napajanja, koncept zaštite, pogonsko iskustvo itd). Koristeći se ovim saznanjima, problem dimenzionalnosti u pretraživanju potencijalnih varijanti rezervnog napajanja je značajno smanjen. U heurističkom pristupu se efikasno kombinuju metode iz ostalih navedenih grupa metoda: estimacija stanja sa proračunom tokova snaga u radijalnim i slaboupetljenim mrežama, kao i metode koje se koriste

za rešavanje problema restauracije, optimizacione metode iz prve grupe i „fuzzy” pristup iz četvrte grupe. Zbog velike efikasnosti, u poslednje vreme ovaj pristup dobija sve više na značaju u restauraciji distributivnih mreža. Nedostatak heurističkog pristupa je što se njegovom primenom ne dobija precizno optimalno rešenje. Međutim, ni kod ostala dva pristupa se ne može garantovati nalaženje optimalnog rešenja.

Treća grupa metoda je bazirana na *ekspertnim sistemima i metodama veštačke inteligencije*. Kod ekspertnih sistema koriste se baze znanja, dok su metode veštačke inteligencije zasnovane ili na neuronskim mrežama ili na genetskim algoritmima. Ova grupa metoda se koristi najčešće za određivanje plana restauracije unutar transformatorske stanice, za dijagnostiku kvara, a ređe i za određivanje sveobuhvatnog plana same restauracije distributivne mreže. Druga značajna primena ovog pristupa je za simulacije i obuku dispečera. Ekspertni sistemi se, zbog velike složenosti problema u distributivnim mrežama, nisu nametnuli u samostalnoj primeni. Oni se najčešće koriste u kombinaciji sa nekom od metoda iz prethodnih grupa.

Veliki broj metoda za restauraciju je zasnovan na teoriji verovatnoće i „fuzzy” pristupu. Ova grupa metoda, *grupa probablističkih metoda i metoda baziranih na „fuzzy” pristupu*, je zasnovana na činjenici da se pri restauraciji opterećenja ne poznaje tačno vrednost opterećenja. Ovo je posebno izraženo pri proceni povratnog opterećenja (pay-back load), koje se javlja pri ponovnom uključivanju potrošača koji su izvesno vreme bili bez napona. Za modelovanje nepoznate vrednosti (povratnog opterećenja) koristi se ili teorija verovatnoće i slučajne promenljive ili „fuzzy” pristup. U slučaju kada se koristi teorija verovatnoća, nepoznata vrednost (povratnog) opterećenja je predstavljena slučajnom promenljivom i njenom raspodelom. U slučaju kada se koristi „fuzzy” pristup, nepoznata vrednost povratnog opterećenja predstavljena je skupom mogućih vrednosti.

Konačno, kod poslednje grupe metoda, *metoda baziranih na tehnikama upravljanja sa rizikom*, suština je da se optimizuje proces donošenja odluka u uslovima neizvesnosti, kada je poznata samo izvesnost (verovatnoća pojave) određenih vrednosti opterećenja.

U ovom programu je predstavljen jedan heuristički višekriterijumski algoritam za određivanje plana restauracije radijalnih distributivnih mreža. Predloženi algoritam se može koristiti u sledećim slučajevima: kvar, remont i preopterećenje napojnog transformatora i izvoda, kao i nefunkcionalnost delovanja relejne zaštite. Od svih gore navedenih problema u ovom zadatku je predstavljen samo kvar na izvodu, kao najreprezentativniji. Ostali problemi se nakon određivanja mreže od interesa suštinski svode na ovaj problem.

Višekriterijumski algoritam se sastoji iz tri globalna koraka. U prvom koraku se dimenzija problema redukuje ograničavanjem razmatranja samo na lokalnu mrežu. Ova mreža predstavlja deo distributivne mreže u okviru koga se ispituje mogućnost restauracije. U drugom koraku se, u okviru tako definisane lokalne mreže, vrši selekcija varijanti rezervnog napajanja korišćenjem algoritma rekonfiguracije na bazi stabla odlučivanja. Ovaj algoritam je izveden iz dobro poznatog

heurističkog algoritma rekonfiguracije na bazi otvaranja rastavljača sa najmanjim strujama. Međutim, za razliku od izvornog algoritma, kojim se kao rešenje dobija jedna bazna radijalna konfiguracija koja najmanje odstupa od optimalne (upetljane) konfiguracije, predloženim algoritmom dobija se proizvoljan broj "kvalitetnih" radijalnih konfiguracija, koje minimalno odstupaju od bazne radijalne konfiguracije. U trećem koraku vrši se rangiranje svih selektovanih varijanti prema korisnički specificiranom kriterijumu. Izbor kriterijuma je u direktnoj zavisnosti od problema koji je restauracijom potrebno prevazići. Konačni izlazni rezultat algoritma je rang lista varijanti rezervnog napajanja distributivne mreže i planova manipulacija za njihovu realizaciju. Pored toga, u okviru algoritma uvažen je i problem neuspelih manipulacija pri realizaciji izabrane varijante.

Tako bi barem trebalo da funkcioniše u teoriji. Ipak, radi jednostavnosti programskog koda, algoritam potreban za rešavanje problema prikazanog u zadatku je pojednostavljen i objašnjen u daljem tekstu.

VIŠEKRITERIJUMSKI ALGORITAM

Višekriterijumski algoritam se sastoji od sledećih globalnih koraka:

1. Određivanje zone „C” – dela mreže u kome se manifestuju problemi.
2. Određivanje lokalne mreže.
3. Selekcija varijanti rezervnog napajanja.
4. Rangiranje varijanti.
5. Problem neuspelih manipulacija.

Zbog kompleksnosti ovog algoritma, a za potrebe razmatranog programskog zadatka, u narednom delu će detaljnije biti opisani samo koraci koji se neophodni za program.

1. **Određivanje zone „C”** – dela mreže u kome se manifestuju problemi.

U ovom koraku algoritma se zadaje tip i lokacija problema. Sledeći tipovi problema se mogu rešiti ovim algoritmom: kvar (remont) i preopterećenje napojnog transformatora i izvoda kao i nefunkcionalnost relejne zaštite izvoda. Nakon zadavanja tipa i lokacije problema, određuje se deo mreže u kome se manifestuju problemi. Kada su u pitanju kvarovi (remonti) mogu se uočiti tri zone: „A”, „B” i „C”.

Zona „A” predstavlja deo mreže od elementa u kvaru (remontu) do napojne transformatorske stanice. Napajanje zone „A” može se jednostavno obnoviti nakon izolacije elementa u kvaru (remontu), zatvaranjem izvodnog prekidača.

Zona „B” je zapravo, element u kvaru (remontu). Ako nije moguće izolovati sam element zbog, na primer, nepostojanja rasklopnih uređaja na krajevima tog elementa, onda se izolovanje

izvodi na nekom od susednih elemenata gde takva oprema postoji, pri čemu i ti elementi ulaze u zonu „B”.

Zona „C” predstavlja deo distributivne mreže koji nakon izolacije kvara, takođe, ostaje bez napona. U slučaju preopterećenja i nefunkcionalnosti ne postoje zone „A” i „B”, dok se zona „C” u tom slučaju odnosi na deo mreže u kojem se problem manifestuje.

- U narednom delu je opisano određivanje zone „C” za sve gore navedene tipove problema.
- *Kvar (remont) napojnog transformatora*: početno stanje za rad algoritma u ovom slučaju je konfiguracija u kojoj je pokvareni transformator isključen; zona „C” se sastoji od svih izvoda koji su se napajali iz razmatranog napojnog transformatora.
 - *Kvar (remont) izvoda*: početno stanje za rad algoritma u ovom slučaju je konfiguracija u kojoj je otvaranjem najbližih rastavljača izolovana sekcija u kvaru; zonu „C” predstavlja preostali deo izvoda koji je nakon izolacije kvara (remonta) ostao bez napona.
 - *Preopterećenje napojnog transformatora*: u ovom slučaju zonu „C” čine izvodi (ili njihovi delovi) koji se napajaju iz tog transformatora i istovremeno imaju normalno otvorene (NO) rastavljače prema izvodima koji se napajaju iz drugih napojnih transformatora.
 - *Preopterećenje izvoda*: u ovom slučaju zonu „C” čini preopterećeni izvod (ili njegov određeni deo).
 - *Nefunkcionalnost delovanja relejne zaštite*: u ovom slučaju postoji određeni deo izvoda koji nije štićen za aktuelno podešenje releja na početku izvoda; ukoliko se drugačijim podešavanjem releja ne može rešiti ovaj problem, neštićeni deo izvoda je potrebno odvojiti od ostatka izvoda i prebaciti ga na neki od susednih izvoda gde će biti adekvatno štićen; taj neštićeni deo izvoda, proširen tako da obuhvati bar jedan NO rastavljač prema susednim izvodima, čini zonu „C”.
- U projektnom zadatku se razmatraju samo prva dva tipa problema.

2. Određivanje lokalne mreže – u ovom koraku se vrši određivanje lokalne mreže. Ova mreža se sastoji od zone „C” i određenog broja izvoda koji imaju mogućnost da zatvaranjem svojih NO rastavljača preuzmu opterećenje zone „C”. Osnovna ideja pri određivanju veličine lokalne mreže je da se brzo i aproksimativnom metodom utvrdi koliki je broj izvoda potreban da se preuzme opterećenje iz zone „C”. U tu svrhu je definisan indeks kapaciteta svakog izvoda kandidata za uključivanje u lokalnu mrežu. Predloženi indeks predstavlja uprošćenu procenu kapaciteta jednog izvoda da preko svog NO rastavljača preuzme opterećenje zone „C”.

Međutim, kao što je već napomenuto, za potrebe zadatka, radi jednostavnosti, rad sa ovim indeksima i formule pomoću kojih se računaju neće biti dalje razmatrani.

3. Selekcija varijanti rezervnog napajanja – iz lokalne mreže mogu se selektovati osnovne i složene varijante rezervnog napajanja. Osnovne varijante su one varijante kod kojih se napajanje zone „C” obezbeđuje sa jednog od susednih izvoda zatvaranjem samo jednog NO rastavljača. Složene varijante su sve one varijante u kojima se napajanje zone „C” obezbeđuje sa više manipulacija nego u osnovnim varijantama (na primer, kada se napajanje zone „C” vrši sa više susednih izvoda, itd).

a) *Određivanje osnovnih varijanti*

Potencijalne osnovne varijante rezervnog napajanja dobijaju se jednostavnim topološkim

pretraživanjem u okviru lokalne mreže. Za svaku takvu potencijalnu osnovnu varijantu trebalo bi da se vrši proračun tokova snaga i provera ograničenja. Razmatraju se sledeća ograničenja: naponska ograničenja u svim čvorovima, termička strujna ograničenja na svim elementima i ograničenja funkcionalnosti relejne zaštite. Tek ukoliko su sva ova ograničenja zadovoljena, takva varijanta smatra se osnovnom varijantom rezervnog napajanja. Ipak, računanje i provera ovih ograničenja bi višestruko povećala kompleksnost zadatka, pa su zbog toga u ovom zadatku zanemarena.

Osnovne varijante su od posebnog značaja iz sledećih razloga:

1. kod njih nema nerestauriranog opterećenja,
2. one se realizuju sa minimalnim troškovima manipulacija,
3. one obezbeđuju kvalitetno rešenje napajanja zone „C” u najvećem broju slučajeva.

b) *Određivanje osnovnih varijanti*

Složene varijante rezervnog napajanja mogu se dobiti:

- I. deobom zone C,
- II. rekonfiguracijom lokalne mreže na bazi stabla odlučivanja, pri čemu se za svaku složenu varijantu, kao i kod prostih varijanti, proračunavaju tokovi snaga i proveravaju ograničenja (strujna, naponska) i funkcionalnost relejne zaštite.

U zadatku se vršila selekcija samo osnovnih varijanti.

Koraci 4. (rangiranje varijanti) i 5. (problem neuspelih manipulacija) se neće dublje razmatrati, jer, iako korisni, nisu neophodni za zadati projektni zadatak.

OPIS PROGRAMA

Program je podeljen u 8 celina, a te celine su fajlovi:

1. *Module_Elements.f90*
2. *Module_Nodes.f90*
3. *Module_Branches.f90*
4. *Module_Load_Grid.f90*
5. *Module_Power_Resoration.f90*
6. *Module_Recursion.f90*
7. *Module_Printer.f90*
8. *Main.f90*

Svi moduli su međusobno povezani upotrebom „Use” komande i to redosledom kako je gore prikazano – *Module_Elements.f90* je povezan u *Module_Nodes.f90* i tako dalje, sve do fajla *Main.f90* koji ih sve zajedno objedinjuje i koristi.

Module_Elements.f90

Ovaj modul definiše klasu *Element* sa svojim poljima i pratećim „get” i „set” procedurama. Polja ove klase su prikazana u tabeli 1.

Ime podatka	Format	Značenje
Element_ID	Integer	Indeks elementa
Element_Status	Logical	Stanje elementa – da li je uključen/isključen
Element_Zone	Character	Pripadnost zoni „A”, „B”, ili „C”
Element_Print	Logical	Govori da li je element bio prikazan u izlaznoj datoteci – neophodno za prikaz različitih varijanti restauracije napajanja

Tabela 1 – polja klase Element

Module_Nodes.f90

Ovaj modul definiše tip *Node* (objekat čvor) sa svojim poljima (prikazanim u tabeli 2), pratećim „get” i „set” procedurama i pripada *Element* nadklasi, pa time nasleđuje njena polja i procedure.

Ime podatka	Format	Značenje
Source_Branch	Integer	Indeks grane sa koje se čvor napaja
Is_Branching	Logical	Grananje čvora – prikazuje da li čvor napaja jednu ili više grana

Tabela 2 – polja objekta Node

Module_Branches.f90

Ovaj modul definiše tip *Branch* (objekat grana) sa svojim poljima (prikazanim u tabeli 3), pratećim „get” i „set” procedurama i pripada *Element* nadklasi, pa time nasleđuje njena polja i procedure.

Ime podatka	Format	Značenje
Start_Node	Integer	Indeks čvora sa kojeg grana „počinje”
End_Node	Integer	Indeks čvora na kom se grana „završava”
Reserve_Branch	Logical	Rezervna grana - da li je grana rezervna za neki od čvorova

Tabela 3 – polja objekta Branch

Module_Load_Grid.f90

Ovaj modul služi za učitavanje podataka iz spoljašnje datoteke *Load_Grid_Data.txt* koji su prikazani u tabeli 4. Učitane podatke upisuje u objekte tipa čvor i grana koje je prethodno kreirao.

Modul koristi promenljive *Element_ID*, *Element_Status*, *Source_Branch*, *Start_Node* i *End_Node*, koje se mogu videti i u klasi *Element* i objektu *Node*. Međutim, te promenljive nisu iste kao one definisane u klasi za elemente i objektu za čvor, nego služe samo za „upisivanje” ulaznih vrednosti iz spoljašnje datoteke i njihovo poklapanje nije namerno.

Ime podatka	Format	Značenje
Number_of_Nodes	Integer	Ukupan broj čvorova
Number_of_Branches	Integer	Ukupan broj grana
Short_Circuit_Location	Integer	Lokacija kratkog spoja
Element_ID	Integer	Identifikacioni broj elementa - indeks
Source_Branch	Integer	Indeks grane koja napaja čvor
Start_Node	Integer	Indeks čvora jednog kraja grane – njenog „početka”
End_Node	Integer	Indeks čvora drugog kraja grane – njenog „završetka”

Tabela 4 –podaci iz spoljašnje datoteke Load_Grid_Data.txt

Module_Power_Resoration.f90

U ovom modulu se nalazi niz procedura i algoritama:

- **Subroutine** *Find_Short_Circuit* – procedura koja pronalazi element čiji se indeks poklapa sa indeksom elementa u kvaru, menja mu status u odričan i zonu u „B”.
 - **Subroutine** *Find_Reserve_Branches* – procedura koja pronalazi grane koje trenutno ne napajaju ni jedan čvor, a u mogućnosti su da to čine. Procedura te grane menja u „rezervne” i postavlja njihov radni status u odričan.
 - **Subroutine** *Find_Branching* – procedura koja pomoću brojača pronalazi čvorove koji su početni čvorovi za dve ili više grana i tim čvorovima stavlja mogućnost grananja na potvrdnu vrednost.
 - **Subroutine** *Restoring_Zone_A* – procedura koja vraća napajanje svim elementima u zoni „A” menjajući im radno stanje na potvrdnu vrednost.
- Kako je određeno koji elementi pripadaju kojim zonama biće objašnjeno u opisu narednog modula.

- **Subroutine** C_Zone_Potentials – procedura koja pronalazi sve potencijalne „napajanje” zone „C”, odnosno grane koje mogu da restauriraju napajanje u toj zoni, pa obeležava te grane sa „X” (menja vrednost njihove zone u „X”).

Module_Recursion.f90

U ovom modulu nalaze se sve rekurzivne metode korišćene u ovom programu:

- **Subroutine** Recursive_Turn_Off – pozivom ove procedure, pozivaju se naredne dve procedure koje služe za isključivanje elemenata koji se u distributivnoj mreži nalaze „iznad” i „ispod” elementa u kvaru čiji je indeks prosleđen u ove dve procedure kao ulazni parametar.
- **Recursive Subroutine** Shutdown_Upstream – procedura prima indeks određenog elementa kao ulazni parametar i isključuje sve elemente koji su povezani sa njim, a nalaze se „iznad” tog elementa. Osim toga, elementi koji su „iznad” elementa u kvaru pripadaju zoni „A”, pa im ova procedura, osim što menja radni status u isključen, dodeljuje i vrednost za polje zone u „A”.
- **Recursive Subroutine** Shutdown_Downstream – slično kao i prethodna procedura, samo što se ova kreće kroz elemente „nadole” u odnosu na prosleđeni element. Elementima koji se nalaze „ispod” zone „B”, biva dodeljena pripadnost zoni „C” i takođe im se menja status u odričnu vrednost.
- **Recursive Subroutine** Zone_C_Reconnect – procedura koja povezuje zonu „C” sa izvorom napajanja. Grana koja ima vrednost zone „X”, povezuje se sa „C” zonom i obezbeđuje joj napajanje. Nakon ovoga, zona te grane prelazi iz „X” u „Y” što označava tu granu kao „u upotrebi”, što je bitno za kasniji prikaz mogućih varijanti restauracije napajanja. Takođe, menja radni status elemenata u zoni „C” u „true” i čvorovima dodeljuje odgovarajuće grane sa kojih se napajaju. Procedura se završava kada prođe kroz sve elemente u zoni „C”.
- **Recursive Subroutine** Zone_C_Disconnect – algoritam sličan prethodnoj proceduri, samo što ova procedura isključuje elemente u „C” zoni (menja im radni status), a grani koja je „C” zonu prethodno napajala (ona sa zonom „Y”), sada biva označena zonom „Z”, čime je sada obeležena kao „iskorištena”. Na ovaj način, kada opet dođe do poziva procedure u novoj varijanti restauracije napajanja, ista grana neće biti opet povezana sa zonom „C” nego će to biti neka druga, označena sa „X”, koja prethodno nije bila iskorištena.
- **Subroutine** Recursive_Print_Branch – procedura koja služi za prikaz šeme mreže u izlaznoj datoteci. Kako se ne bi dešavalo da se jedni te isti elementi prikazuju više puta, procedura tim elementima menja atribut Element_Print u „true”.
- **Subroutine** Recursive_Print_Node – slično kao prethodna procedura sa kojom zajedno radi na prikazu šeme u izlaznu datoteku, ova procedura prikazuje čvorove. Ove

dve procedure se međusobno pozivaju u zavisnosti od toga do kog elementa je algoritam „došao”.

- **Recursive Subroutine** Branching_Print – procedura čija je namena da prikaže grananje unutar distributivne mreže u izlaznu datoteku. Radi slično kao prethodne dve procedure.

Module_Printer.f90

U ovom modulu se pozivaju sve procedure za prikaz rešenja zadatka u izlaznu datoteku:

- **Subroutine** Print_Zone_A – procedura koja proverava da li u mreži postoje elementi u zoni „A”. Ukoliko postoje, poziva se procedura Restoring_Zone_A iz Module_Power_Restoration.f90
 - **Subroutine** Print_Zone_C – kao i prethodna, proverava da li postoje elementi u zoni „C” i ukoliko postoje, poziva procedure Find_Reserve_Branches i C_Zone_Potentials iz Module_Power_Restoration.f90, kao i proceduru Variations_of_C_Zone_Restoration čije se objašnjenje nalazi ispod.
 - **Subroutine** Variations_of_C_Zone_Restoration – procedura prikazuje u izlaznu datoteku *Result_Output.txt* varijacije distributivne mreže nakon svake moguće restauracije napajanja za zonu „C”. Procedura prvo traži grane sa zonom „X”, te poziva proceduru Zone_C_Reconnect iz modula Recursion, proceduru Print_Output i proceduru Zone_C_Disconnect kako bi se elementi u zoni „C” opet isključili i spremili za novu varijantu povezivanja.
 - **Subroutine** Print_Output – procedura koja ispisuje konačan rezultat u izlaznu datoteku pozivajući sve ostale „Print” procedure poput Recursive_Print_Branch i Branching_Print iz Recursion modula, nakon čega svim elementima vraća vrednost polja Element_Print na „false” čime ih sprema za ispis nove varijante restauracije napajanja.
 - **Subroutine** Print_Branch – procedura koja u terminal ispisuje sve podatke vezane za objekat grane.
 - **Subroutine** Print_Node – procedura koja u terminal ispisuje sve podatke vezane za objekat čvora.
 - **Subroutine** Print_Table – procedura koja objedinjuje prethodne dve, poziva ih i prikazuje podatke svih elemenata u obliku tabele u terminalu.
- Ove tri procedure služe isključivo radi lakšeg praćenja koraka koje preduzima program i radi jednostavnijeg traženja mogućih grešaka.

Main.f90

Glavni deo programa, povezan sa svim modulima programa. U njemu se nalaze komande koje ispisuju u terminal odrađene korake programa, radi lakšeg praćenja rada programa. Ovaj deo programa poziva procedure `Load_Grid_Data` koja je definisana u moduli `Load_Grid`, `Find_Short_Circuit` i `Find_Branching` definisane u modulu `Power_restoration`, `Recursive_Turn_Off` iz modula `Recursion`, otvara spoljašnju datoteku namenjenu za ispis rezultata (*Result_Output.txt*), zatim poziva `Print_Zone_A` i `Print_Zone_C` iz modula `Printer`, zatvara spoljašnju datoteku i na kraju dealocira memoriju koja je bila rezervisana za objekte tipa čvor i grana na početku programa. Ovde se program završava.