Cuprins

[1 Specificații editare pentru lucrarea de finalizare a studiilor 5](#_Toc532486141)

[1.1 Subtitlu 5](#_Toc532486142)

[1.1.1 Sub-subtitlu 5](#_Toc532486143)

[1.1.2 Text 5](#_Toc532486144)

[1.1.3 Ecuații, tabele, figuri 5](#_Toc532486145)

[1.1.4 Editare bibliografie 6](#_Toc532486146)

[2 Acesta este titlul capitolului 2 7](#_Toc532486147)

[2.1 Subtitlu din capitolul 2 7](#_Toc532486148)

[2.1.1 Sub-subtitlu din capitolul 2 7](#_Toc532486149)

[2.2 Subtitlu din capitolul 2 8](#_Toc532486150)

[2.2.1 Sub-subtitlu din capitolul 2 8](#_Toc532486151)

Observație: cuprinsul se actualizează automat după selecția întregului document (CTRL+A) li apăsarea tastei F9 !!!!! → după actualizare ștergeți această observație având grijă ca formatarea să rămână pe stilul “Body Text First Indent”

Diploma project summary

Stilul “Bibliography” este folosit aici pentru titlul rezumatului în limba engleză. Acest stil este util în orice situație când se dorește inserarea unui titlul nenumerotat.

Completați această secțiune folosind aceleași stiluri de editare ca și în corpul lucrării. Template-ul generează automat un page-break și trece la secțiunea următoare la finalul ultimului aliniat.

Planificarea activității

Se completează în mod similar ca și secțiunea cu rezumatul în limba engleză.

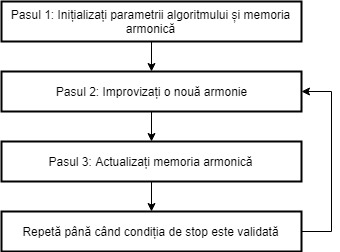
* Modificarea rezistenței electrice cu umiditatea ale unor rezistoare cu ceramică sau macromolecule organice
* Modificarea impedanței cu umiditatea a unui electrolit cu polimer
* Modificarea capacității electrice cu umiditatea a unor condensatoare realizate în tehnologia filmelor subțiri

Stadiul actual

Microrețelele reprezintă cea mai recentă soluție utilizată pentru creșterea autosustenabilității și fiabilității viitoarelor rețele electrice de distribuție. O microrețea izolată dispune de surse de energie regenerabile, un sistem de stocare a energiei și consumatori. Asigurarea unui management al energiei adecvat pentru microrețele este o problemă dificilă. Există abordări diferite pentru rezolvarea acestei probleme. În acest capitol se vor prezenta trei algoritmi folosiți în managementul energiei pentru microrețele, respectiv câteva idei despre algoritmul folosit pentru realizarea lucrării de diplomă.

Primul algoritm folosit pentru rezolvarea problemei planificării zilnice este Harmony Search Optimization. Acest procedeu a fost dezvoltat de Geem și colaboratorii în [1], având aceeași structură de bază ca și alți algoritmi metaeuristici de căutare, cu caracteristici cum ar fi păstrarea istoriei vectorilor anteriori, variind rata lor de adaptare în timpul procesului de căutare (similar algoritmului Simulated Annealing) și luând în considerare mai mulți vectori simultan. Ceea ce îl face diferit este noua metodă de generare a soluției: noul vector este generat de toți vectorii existenți, algoritmul nu necesită modele de probabilitate pentru a estima o distribuire a unor noi valori promițătoare și noile valori sunt selectate din memorie, precum și eventualele intervale de valori [2].

Harmony Search a fost inspirat din procesul în care muzicienii o utilizează pentru a obține o armonie plăcută. În acest caz, calitatea estetică a muzicii este înlocuită cu funcția obiectiv. Deoarece pitch-ul fiecărui instrument influențează estetica generală a muzicii, ele sunt reprezentate de variabilele de decizie a problemei de optimizare, formând împreună vectorul de soluție. În procesul lor de improvizație muzicienii au trei opțiuni: să folosească un pitch din memorie, să adapteze ușor un pitch din memorie sau să utilizeze un pitch aleatoriu din intervalul posibil. Prin urmare, în procesul de optimizare, o variabilă de decizie poate primi o valoare din memoria armonică, poate primi o valoare din memoria armonică care a fost ușor ajustată sau poate primi o valoare aleatorie din intervalul posibil definit la începutul algoritmului de optimizare. Există trei parametri care influențează performanța algoritmului: rata de luare în considerare a armoniei, rata de ajustare a pitch-ului și lățimea de bandă. Probabilitatea alegerii unei armonii din memoria armonică este dată de către rata de luare în considerare a armoniei. Rata de ajustare a pitch-ului reprezintă probabilitatea ajustării unei armonii din memoria armonică. Lățimea de bandă este folosit pentru a obține un echilibru între un proces local de optimizare și unul global. După ce variabilele de decizie primesc noi valori și se formează un nou vector de soluție posibil, se evaluează folosind funcția obiectiv. Dacă noul vector de soluție îndeplinește toate constrângerile și este mai bun decât cel mai rău vector de soluție din memoria armonică, atunci acesta din urmă va fi înlocuit. În Figura 1 sunt prezentați pașii folosiți în algoritmul Harmony Search Optimization. Algoritmul se termină atunci când se validează condiția de stop, deobicei fiind numărul maxim de iterații.

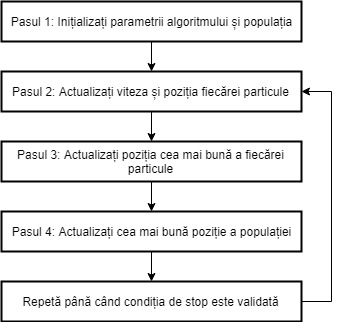


Figură 1. Pașii utilizați de algoritmul Harmony Search Optimization

Rezultatele obținute prin intermediul acestui algoritm arată că este o soluție viabilă pentru programarea zilnică a unei microrețele izolate. Cererea de sarcină este satisfăcută de energiile regenerabile utilizate și starea de încărcare din baterii este în limitele predefinite. În plus, se încarcă bateriile disponibile fără a produce o cantitate mai mare de energie care nu poate fi stocată.

A doua procedură utilizată este Particle Swarm Optimization. Dezvoltat de Kennedy și Eberhart în [3], Particle Swarm Optimization este o căutare stocastică bazată pe populație, fiind inspirat din comportamentul natural al unui stol de păsări sau al unui banc de pești pentru căutarea hranei lor. Similar cu alți algoritmi de căutare metaeuristici, Particle Swarm Optimization începe cu o populație generată aleatoriu de soluții posibile și converge în mod ideal spre soluția optimă globală a problemei. Spre deosebire de alte tehnici de optimizare, algoritmul Particle Swarm Optimization adaptează fiecare dintre soluțiile sale posibile bazate pe cel mai bun rezultat al acestora, precum și pe cel mai bun rezultat general al întregii populații. Parametrii utilizați pentru generarea de noi soluții variază de la o versiune a algoritmului la alta [4].

Algoritmul Particle Swarm Optimization încearcă să imite comportamentul animalelor în căutarea lor pentru hrană. Stolul de păsări sau bancul de pești este reprezentat de către populație, fiecare particulă preluând rolul unui individ. Funcția obiectiv reprezintă sursa de hrană. Dimensiunea problemei este dată de către numărul de variabile de decizie. În căutarea hranei animalele își adaptează poziția atât individual, cât și colectiv. Prin urmare, în procesul de optimizare, atât cea mai bună poziție a particulei cât și cea mai bună poziție a întregii populații sunt luate în considerare atunci când se generează noi soluții posibile. Algoritmul începe cu o mulțime inițială de particule generate aleatoriu în spațiul de căutare. Particulele sunt inițializate fiecare cu o poziție aleatorie și o viteză aleatorie. Pe măsură ce se mișcă populația, fiecare particulă este actualizată cu o nouă poziție și o nouă viteză. Viteza nouă a particulelor se calculează pe baza vitezei sale curente, a celei mai bune poziții și a celei mai bune poziții a mulțimii. Poziția nouă se calculează utilizând poziția curentă și viteza nouă calculată. Când se calculează noua viteză și poziție pentru fiecare particulă se utilizează trei parametri: greutatea inerției, greutatea cognitivă și greutatea socială. Greutatea inerției reprezintă cantitatea de influență pe care o are viteza curentă asupra celei următoare. Greutatea cognitivă și greutatea socială reprezintă influența pe care poziția cea mai bună a particulei și poziția cea mai cunoscută a mulțimi o au în calcularea noii poziții a particulelor. Cea mai bună poziție atinsă de fiecare particulă, precum și de întreaga mulțime sunt actualizate cu fiecare iterație. În Figura 2 putem observa pașii utilizați de către algoritm. Deobicei, când se atinge numărul maxim de iterații, algoritmul Particle Swarm Optimization se termină.



Figură 2. Pașii utilizați de algoritmul Particle Swarm Optimization

Rezultatele obținute prin utilizarea algoritmului Particle Swarm Optimization arată că folosirea acesteia pentru programarea zilnică a unei microrețele izolate este o soluție viabilă. Nu numai că rezultatele sunt comparabile cu cele obținute folosind alți algoritmi de optimizare metaeuristică, dar și timpul de calcul este mai scurt. Elementele funcției microrețelei în cadrul parametrilor definiți și cerințele energetice sunt asigurate de surse regenerabile de energie.

Al treilea procedeu folosit pentru rezolvarea acestei probleme este algoritmul genetic. Algoritmii genetici sunt tehnici adaptive de căutare euristică, bazate pe principiile geneticii și ale selecției naturale, enunțate de Darwin (“supaviețuiește cel care e cel mai bine adaptat”). Mecanismul este similar procesului biologic al evoluției. Acest proces posedă o trăsătură prin care numai speciile care se adaptează mai bine la mediu sunt capabile să supraviețuiască și să evolueze peste generații, în timp ce acelea mai puțin adaptate nu reușesc să supraviețuiască și cu timpul dispar, ca urmare a selecției naturale. Probabilitatea ca specia să supraviețuiască și să evolueze peste generații devine cu atât mai mare cu cât gradul de adaptare crește, ceea ce în termeni de optimizare înseamnă că soluția se apropie de optim.

Un algoritm genetic este un model informatic care emulează modelul biologic evoluționist pentru a rezolva probleme de optimizare ori căutare. Acesta cuprinde un set de elemente individuale reprezentate sub forma unor șiruri binare (populația) și un set de operatori de natură biologică definiți asupra populației. Cu ajutorul operatorilor, algoritmii genetici manipulează cele mai promițătoare șiruri, evaluate conform unei funcții obiectiv, căutând soluții mai bune. Algoritmii genetici au început să fie recunoscuți ca tehnici de optimizare odată cu lucrările lui John Holland.

Ca aplicații practice, algoritmii genetici sunt cel mai adesea utilizați în rezolvarea problemelor de optimizare, planificare ori căutare. Condiția esențială pentru succesul unei aplicații cu agenți inteligenți este ca problema de rezolvat să nu ceară obținerea soluției optime, ci să fie suficientă și o soluție apropiată de optim.

A patra metodă folosită în programarea zilnică a unei microrețele izolate este algoritmul Simulated Annealing, fiind și procedeul folosit pentru realizarea acestei lucrări de diplomă. Este o tehnică probabilistică pentru aproximarea optimului global al unei funcții date. Mai exact, este o metodă euristică de a aproxima optimizarea globală într-un spațiu de căutare mare pentru o problemă de optimizare. Este adesea folosit atunci când spațiul de căutare este discret. Pentru problemele în care găsirea unui optim global aproximativ este mai importantă decât găsirea unui optim local precis într-o anumită perioadă de timp, Simulated Annealing poate fi preferabilă alternativelor, cum ar fi coborârea gradientului.

Numele și inspirația provin din călirea în metalurgie, o tehnică care implică încălzirea și răcirea controlată a unui material pentru a mări dimensiunea cristalelor sale și a reduce defectele lor. Ambele sunt atribuite materialului, depinzând de energia liberă termodinamică. Încălzirea și răcirea materialului afectează atât temperatura cât și energia liberă termodinamică. Acest algoritm poate fi folosit pentru a găsi o aproximare a unui minim global pentru o funcție cu un număr mare de variabile. Această noțiune de răcire lentă implementată în Simulated Annealing este interpretată ca o scădere lentă a probabilității de acceptare a soluțiilor mai slabe pe măsură ce spațiul soluției este explorat. Acceptarea soluțiilor mai slabe este o proprietate fundamentală a metaeuristici, deoarece permite o căutare mai amplă a soluției optime globale. În general, algoritmii Simualted Annealing funcționează după cum urmează: la fiecare etapă, algoritmul selectează aleator o soluție apropiată de cea actuală, măsoară calitatea sa și apoi decide să se mute la ea sau să rămână cu soluția actuală pe baza oricăreia dintre cele două probabilități între care alege pe baza faptului că noua soluție este mai bună sau mai rea decât cea actuală. În timpul căutării, temperatura este scăzută treptat de la o valoare pozitivă inițială la zero și afectează cele două probabilități: la fiecare etapă, probabilitatea de a trece la o soluție mai bună este fie menținută la 1, fie este schimbată spre o valoare pozitivă. Pe de altă parte, probabilitatea de a trece la o soluție mai rea nouă este treptat schimbată spre zero.

# Fundamentare teoretică

## Microrețeaua

### Introducere

O microrețea este un sistem electroenergetic de dimensiuni mici ce include una sau mai multe grupuri de generare distribuită care pot funcționa independent de sistemul electroenergetic național sau conectat cu acesta.

În mod clasic, energia electrică este transferată dinspre rețeaua de transport spre rețeaua de distribuție, astfel că se produc pierderi de putere pe lanțul de elemente de rețea, respectiv printr-un număr mare de linii (exprimat în km) și transformatoare. Dezvoltarea surselor de putere mică, funcționând fie pe combustibil convențional, fie folosind surse regenerabile, a permis creșterea numărului de generatoare prezente în rețeaua de distribuție. Existența unor surse de energie electrică într-o zonă de rețea de distribuție a condus la ideea de arhitectură de microrețea. O microrețea se utilizează deoarece:

* Costul energiei se poate micșora (comparativ cu energia primită din sistemul electroenergetic principal)
* Fiabilitatea și calitatea energiei se pot îmbunătăți
* Poate crește eficiența și se reduc emisiile de noxe
* Poate fi singura opțiune dacă infrastructura de transport nouă sau modernizată nu poate fi dezvoltată într-un timp sau cu un cost eficace

### Scurt istoric

În perioada de început a dezvoltării sistemelor electroenergetice (1880 – 1910), acestea aveau o structură simplă de microrețea. O singură centrală electrică alimenta o zonă de consum. Ulterior, ideea de funcționare izolată, sub forma unei microrețele, a început să dispară în perioada 1910 – 1950 prin politica de interconectare a rețelelor electrice, datorită avantajelor pe care acestea le prezintă. În prezent, ideea de microrețea a început să prezinte un nou sens ca urmare a dezvoltării de noi tehnologii, existența restricțiilor privind construirea de noi elemente în rețelele de transport și distribuție, aspectelor legate de mediu, cerințelor legate de fiabilitate.

Societatea modernă se bazează pe un sistem de furnizare a energiei electrice de înaltă fiabilitate. Problemele actuale privind disponibilitatea energiei primare, îmbătrânirea infrastructurii de transport și distribuție a rețelelor electrice, necesitatea de a instala surse noi de producție (cum ar fi sursele regenerabile) și vânzarea energiei electrice prin piețele angro constituie o provocare pentru operatorii de sistem în ceea ce privește securitatea, siguranța și calitatea. De aceea sunt necesare investiții importante pentru dezvoltarea și modernizarea infrastructurii electrice, iar cel mai eficient mod pentru a răspunde cerințelor sociale va fi încorporarea de soluții inovatoare, tehnologii și arhitecturi de rețele.

Viitoarele rețele electrice vor trebui să se adapteze la schimbările tehnologice, să răspundă valorilor societății privind mediul înconjurător și aspectul comercial. Astfel, securitatea sistemului, siguranța, mediul înconjurător, calitatea energiei electrice, costul furnizării și eficiența energetică sunt evaluate într-o manieră nouă, ca răspuns la schimbarea cerințelor de pe piață. Tehnologia trebuie să demonstreze siguranță, rezistență și eficiență. La nivelul distribuției, noile cerințe pentru dezvoltare sunt:

* Rețelele de distribuție trebuie să prezinte accesibilitate pentru producerea distribuită (GD) și sursele de energie regenerabile (SER), care pot fi auto-dispecerizate sau dispecerizate prin intermediul unui dispecer local
* Rețelele de distribuție care permit managementul consumului local și care interacționează cu utilizatorii finali prin sisteme de contorizare inteligente

Rețelele de distribuție încep să se transforme din rețele pasive în rețele active în sensul în care luarea unei decizii și controlul sunt distribuite, iar puterea circulă bidirecțional. Acest tip de rețele cu participarea generării distribuite, a surselor de energie regenerabile și a dispozitivelor de stocare, oferă soluții pentru noi tipuri de echipamente și servicii, fiecare din acestea fiind necesar să respecte standardele și protocoalele comune. Funcția unei rețele de distribuție active este să interconecteze în mod eficient sursele de putere și consumatorii, permițândule ambelor părți să decidă care este cel mai bun mod de funcționare în timp real. Determinarea circulației de puteri, controlul tensiunii și sistemele de protecții necesită tehnologii de cost competitiv și sisteme de comunicare noi, care încorporează tehnologii comunicaționale și informatice.

Crearea rețelelor de distribuție active vor permite în mod radical dezvoltarea de noi concepte. Probabil că cel mai promițător concept este acela de microrețele. Microrețelele cuprind sisteme de distribuție de joasă tensiune (LV) cu surse distribuite de energie (DERs) cum ar fi microturbinele, pile cu combustibil, celule fotoelectrice, precum și dispozitive de stocare (baterii cu condensatoare de dimensiuni mari, baterii de stocare de dimensiuni mici), respectiv consumatori controlabili, oferind posibilități de control considerabile în ceea ce privește funcționarea rețelei electrice.

Aceste sisteme sunt conectate la rețeaua de distribuție de medie tensiune, dar pot funcționa și izolat de rețeaua principală în cazul unor defecte în rețea. Din punctul de vedere al consumatorilor, microrețelele asigură atât energie termică cât și energie electrică, iar în plus îmbunătățesc siguranța în funcționare la nivel local, reduc emisiile de nox, îmbunătățesc calitatea energiei prin controlul tensiunii și reducerea golurilor de tensiune, pot oferi costuri reduse pentru furnizarea energiei electrice sau termice. Din punctul de vedere al rețelelor, o microrețea poate fi privită ca o entitate controlabilă în cadrul sistemului energetic care poate să funcționeze ca o mică sursă de putere sau ca servicii auxiliare, sprijinind rețeaua principală.

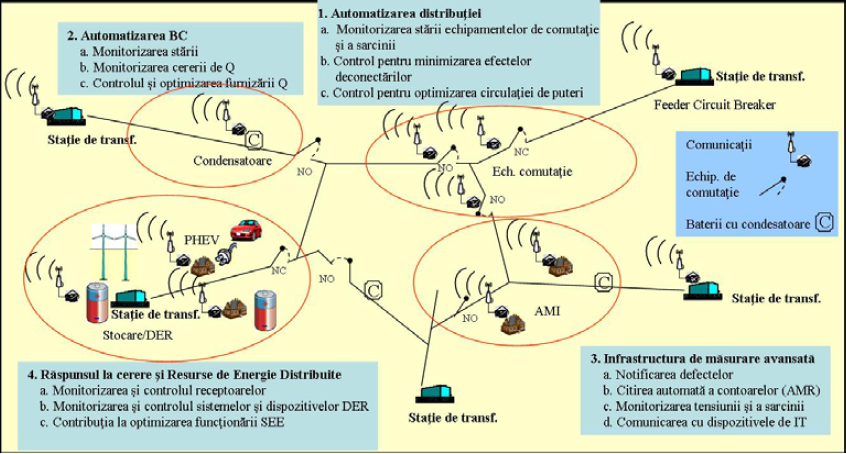
Potențialul economic cheie al aplicațiilor GD la premisele de client constă în oportunitatea de utilizare locală a căldurii evacuate de la conversia combustibilului primar în electricitate. În ultimii ani au existat progrese remarcabile privind dezvoltarea de aplicații de cogenerare de mică putere. Aceste sisteme se așteaptă să joace un rol important în microrețelele din tările nordice. Pe de altă parte, sistemele PV se anticipează că se vor dezvolta semnifiativ în țările cu climat însorit. Aplicații de microcogenerare și PV au un potențial de creștere a eficienței totale a utilizării surselor de energie primară și în consecință, reduce substanțial emisiile de carbon și noxe, constituind un alt beneficiu important în eforturile omenirii de a combate schimbarea climatică.

Din punctul de vedere al rețelei electrice, aplicațiile GD pot reduce necesitatea de noi investiții în rețelele electrice de distribuție și de transport. Generatoarele distribuite amplasate în apropierea consumatorilor vor decongestiona căile de alimentare și vor conduce la reducerea pierderilor de putere și posibilitatea de rezervare a unor elemente din rețelele principale. Microrețelele pot, de asemenea, să asigure servicii de sistem în cazul unor probleme ce apar în rețeaua electrică principală.

În consecință, microrețelele sunt subiectul unei cercetări intense și activități de propagare în America, Japonia, Europa, Canada pentru a furniza soluțiile eficiente și a demonstra conceptele de funcționare ale microrețelelor.

### Caracteristici ale unei microrețele

* Valoare de vârf a consumului de energie electrică: 1kW – 100 MW
* Consum de energie termică: 0.5 MJ/h – 1000 MJ/h
* Numărul de consumatori alimentați: 1 – 50.000
* Tipul consumatorilor: rezidențial, comercial sau industrial
* Întinderea geografică: de la o casă până la 10 kmp
* Funcționarea mixtă: microrețeaua poate fi configurată pentru a comuta între funcționarea “izolată” și “ne-izolată” în funcție de starea rețelei publice
* Funcționarea izolată: microrețeaua funcționează independent de rețeaua publică
* Nivelul tensiunii: JT sau MT, AC sau DC
* Arhitectura: radială sau buclată cu una sau mai multe generatoare

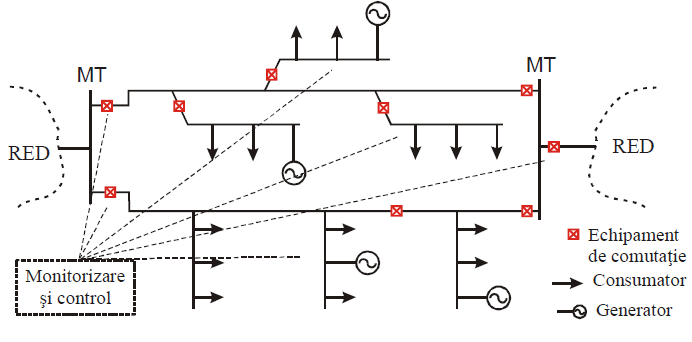


Figură 3. Funcțiile unei microrețele

Pe lângă infrastructura clasică a unei rețele electrice, ce cuprinde linii, transformatoare, sisteme de protecții și automatizări, o microrețea include consumatori inteligenți generatoare distribuite, sisteme de detectare a defectelor avansate, echipamente de comutație inteligente, o infrastructură de măsurare avansată, căi de alimentare de rezervă, precum și un sistem de monitorizare și control ce cuprinde produse informatice și cu ajutorul cărora se pot îndeplini următoarele obiective:

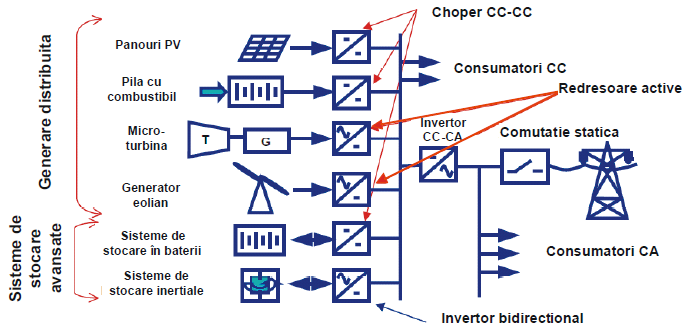
* Izolarea microrețelei în cazul producerii unui blackout în SEN
* Optimizarea costului energiei electrice, comparativ cu cel obținut pe piața de energie electrică, folosind resursele proprii, atât generatoarele electrice și sursele de stocare, precum și consumatorii activi
* Îmbunătățirea fiabilității și calității energiei electrice prin posibilitatea de reconfigurare a rețelei electrie
* Creșterea eficienței și reducerea emisiilor de substanțe poluante prin integrarea surselor regenerabile de energie, precum panourile fotovoltaice, micro-hidrocentralele, turbinele eoliene

O microrețea se poate dezvolta în cadrul unei rețele electrice de distribuție (RED), de medie tensiune sau de joasă tensiune.



Figură 4. Reprezentarea de bază a unei microrețele într-o rețea de distribuție

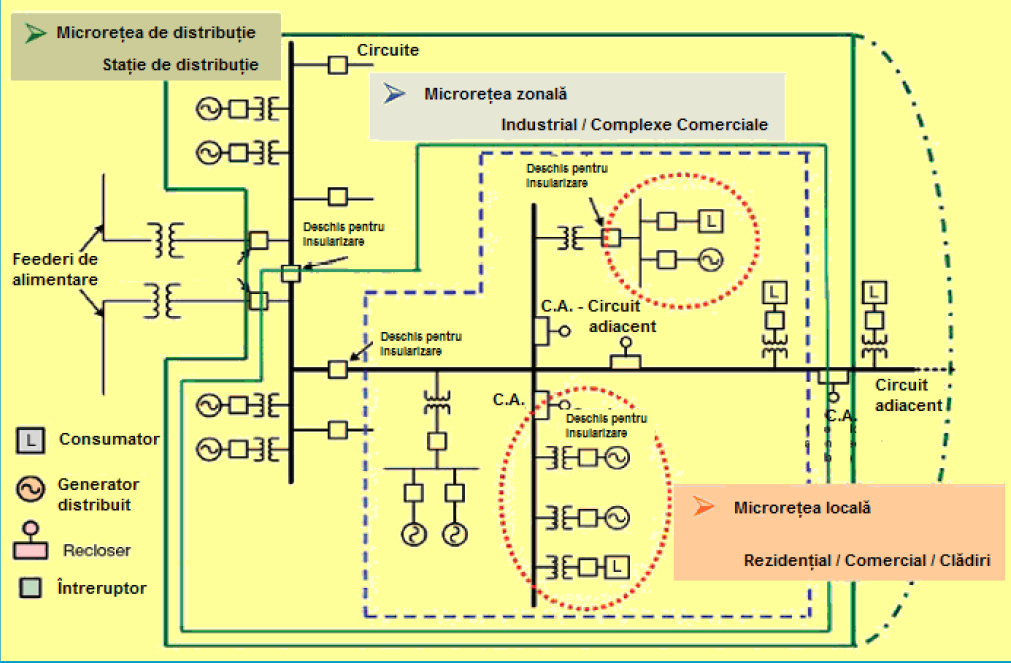
În cazul zonelor izolate, microrețelele pot constitui singura opțiune din punct de vedere tehnic sau economic. În plus, alimentarea cu energie electrică la tensiune continuă poate reprezenta o soluție mult mai eficientă decât alimentarea la tensiune alternativă. O atenție deosebită trebuie acordată tensiunii continue și electronicii de putere pentru conectarea la rețeaua electrică a unor surse care produc energie electrică la tensiune continuă (C.C).



Figură 5. Conectarea la rețeaua electrică a surselor de energie electrică

### Arhitectura Microrețelelor

O microrețea poate cuprinde o parte din sistemele de distribuție de MT și JT și consumatori agregați deserviți de unul sau mai multe unități de GD. Din punct de vedere al operaționalului, o microrețea poate funcționa interconectat cu sistemul electroenergetic principal prin intermediul unui punct comun de conectare (PCC), având astfel posibilitatea de a se izola în cazul unor defecte ce apar în rețeaua electrică de distribuție sau transport.



Figură 6. Ilustrarea microrețelelor

În timp ce este fizic conectată la rețeaua principală, modul de operare și control al microrețelei poate să se schimbe de la mod dependent de rețeaua principală la mod independent de rețeaua principală, în funcție de puterea interschimbată între microrețea și rețeaua electrică principală.

În Figura 6 se prezintă sugestiv 3 tipuri de microrețele și anume: microrețea de distribuție / industrială (care aparține unei companii de distribuție), microrețea zonală cu utilitate singular / multifuncțională de tip industrial sau comercial și microrețea locală destinată sectorului rezidențial sau clădirilor mici.

#### Microrețele de distribuție

Accesul la microrețele poate facilita pe scară largă utilizarea SRE și / sau încorpora sisteme de cogenerare (CHP) în rețeaua de distribuție, amortizând în același timp fluctuațiile de energie ale rețelei principale. Microrețelele pot fi formate dintr-o parte sau de către toți fiderii care pleacă dintr-o stație de distribuție, fiind condusă de un operator de distribuție. Utilizarea unui număr mare de generatoare distribuite localizate aproape de centrele de consum, o microrețea industrială poate asigura echilibrul local între puterile active generate și consumate, respectiv pot contribui la evitarea unor congestii pe fiderii de medie tensiune. La nivelul companiei de distribuție ce are în proprietate microrețeaua, turbine hidro de putere mică, celule fotoelectrice, grupurile eoliene și sistemele de biomasă sunt câteva din sursele de energie alternativă care pot fi utilizate cu emisii reduse de noxe. O microrețea de utilitate publică poate să fie deconectată de la rețeaua principală în timpul perioadelor programate de întreținere a fiderilor de înaltă tensiune și stațiilor într-un mod controlat. Izolarea planificată a microrețelei limitează durata întreruperilor în alimentarea consumatorilor. O microrețea de utilitate publică poate să ofere și servicii de sistem cum ar fi furnizarea de energie reactivă și creșterea calității energiei electrice.

De asemenea, câteva tehnologii GD pot asigura putere reactivă dispecerizabilă care să compenseze puterea reactivă a sarcinii locale și să mențină nivelul de tensiune. Folosirea surselor de cogenerare (CHP), microrețeaua de utilitate publică poate să ofere energie termică din procesul de producere al energiei electrice sub formă de căldură sau apă fierbinte (sau abur) pentru uz casnic. Conceptul de CHP în cadrul microrețelei este aplicat prin amplasarea optimă a surselor CHP acolo unde echipamentele termice / electrice cresc complet eficiența instalației și reduc consumul de combustibil.

#### Microrețele zonale – industriale și comerciale

Consumatorii comerciali și industriali de energie electrică sunt definiți prin clase de importanță, care se referă la gradul de calitate al energiei electrice. Un consumator critic poate să nu tolereze întreruperile în alimentarea cu energie electrică. O microrețea poate fi folosită cu ușurință pentru a alimenta mai multe tipuri de consumatori industriali sau comerciali, de exemplu: campus universitar, centru comercial, instalații industriale etc. Strategiile de management avansat al sarcinii din cadrul microrețelelor constau într-un control distribuit și automat care să prevină întreruperile neplanificate și deci să contribuie la îmbunătățirea calității energiei electrice prin creșterea independenței microrețelei de rețeaua electrică principală.

Utilizarea conceptului de microrețea, cu un nivel distinct de fiabilitate și calitate a energiei electrice poate fi definit pe baza unei clasificări a consumatorilor și diferențierea serviciilor pentru utilizatori multipli. Clasificarea consumatorilor și controlul acestora în cadrul unei microrețele poate, de asemenea, contribui la aplatisarea curbei de sarcină fie în mod izolat, fie în mod conectat la rețea. O microrețea comercială sau industrială se poate izola atunci când calitatea energiei electrice din rețeaua principală nu satisface cerințele impuse și poate afecta inclusiv calitatea energiei electrice asigurate de microrețea. Funcționarea independentă față de rețeaua principală a unei microrețele comerciale / industriale poate fi planificată, de exemplu, la vârf de sarcină când prețul energiei electrice absorbite din rețeaua principală este ridicat. O microrețea poate alimenta un consumator rezidențial mic, adică un grup de case orășenești. Microrețeaua rezidențială asigură un sistem de furnizare a energiei electrice convenabilă și eficientă, fiind particularizată în funcție de cerințele consumatorilor și generatoarelor distribuite utilizate. Generația de panouri solare și microturbine în cogenerare constituie surse distributive atractive pentru aplicațiile rezidențiale și clădirile comerciale. Sursele PV pot fi încorporate în structura clădirii. Proprietarii clădirii pot beneficia de bună corelare între vârful curbei de sarcină și intensitatea soarelui pentru producerea de energie solară. Modulele la scară redusă de microturbine oferă surse de cogenerare controlabile și eficiente de energie electrică și căldură, cu zgomot mic care pot fi instalate individual în apartamente sau birouri, acolo unde consumul de energie electrică se realizează la o eficiență ridicată. Bazându-ne pe accesul la microrețea, consumul de energie termică și electrică total al rețelelor locale este controlat prin putere adecvată și strategii de management, cu rol de a personaliza prețul energiei și de a limita impactul fluctuațiilor de energie ale surselor intermitente și schimbăriile bruște ale consumului în rețea.

#### Microrețele locale

Electrificarea comunităților retrase și a zonelor neîntegrate în țările dezvoltate și a insulelor geografice este o mare prioritate pentur companiile de utilități din toată lumea. Câteva țări au cercetat adoptarea unui concept de producere de energie descentralizată, microrețeaua, pentru furnizarea de energie în zone izolate. Cererile de energie ale zonelor neîntegrate pot fi satisfăcute prin instalarea regenerabilelor și alternativelor DER pentru a forma rețele izolate și microrețele autonome care furnizează energie electrică și eventual căldură sau apă caldă clienților locali sau comerciali.

În funcție de caracteristicile geografice ale unei zone retrase și disponibilitatea resurselor de diverse tipuri cum ar fi: microturbine, mori de vânt, celule fotovoltaice și turbine pe gaz cu emisii mici, pot fi folosite. O deosebire majoră în modelul microrețelelor depărtate este aceea că producerea trebuie fie dimensionată pentru a servi întregii saricini cu un nivel adecvat al capacității de rezervă. În plus, dispersarea încărcării și marile diferențe între încărcarea minimă și maximă a microrețelei fac din tehnologia selecției, a mărimii DER, un lucru competitiv. Următoarele metode sunt sugerate pentru a realiza balanțul de energie pe termen scurt sau pe termen lung a microrețelelor retrase menite să învingă fluctuațiile de putere introduse de producerea intermitentă și sarcinii variabile:

* Participare avansată a puterii și angajamentul unităților printr-un set de mai multe surse de generare pentru a selecta combinația potrivită a DER în funcție de variația sarcinii
* Utilizarea mărimii optime a unităților de energie
* Control avansat al sarcinii

### Moduri de funcționare a microrețelelor

#### Modul conectat la rețeaua electrică publică

Acest mod de funcționare este un mod normal deoarece, pentru asigurarea unor parametrii electrici în conformitate cu cerințele de calitate a energiei electrice, este necesară funcționarea interconectată.

Din punct de vedere al frecvenței, aceasta poate fi menținută foarte aproape de valoarea nominală de 50 Hz, doar în condițiile funcționării interconectate.

Funcționarea interconectată asigură condițiile necesare accesului la cele mai ieftine surse de energie electrică. Consumatorii din microrețea pot să opteze pentru a folosi sursele de energie electrică din cadrul microrețelei sau pentru a cumpăra energie electrică din rețeaua publică.

#### Modul izolat, cu funcționare independentă de rețeaua electrică publică

În cazul în care în rețeaua publică se produce un incident care conduce la întreruperea alimentării cu energie electrică, microrețeaua trebuie să fie capabilă să asigure alimentarea de rezervă a principalilor consumatori de energie electrică localizați în cadrul microrețelei. Funcționarea corespunzătoare în aceste condiții presupune existența unui dispecer local, dar și a unor sisteme de reglaj pentru funcționarea izolată.

## Algoritmul Simulated Annealing

### Introducere

Simulated Annealing este o metodă probabilistică propusă de către Kirkpatrick, Gellet, Vecchi (1983) și Cerny (1985). Ideea acestui algoritm provine de la analogia dintre căutarea soluției unei probleme de optimizare și evoluția stărilor unui solid supus unui tratament termic care începe printr-o încălzire bruscă și continuă cu o răcire lentă. La temperatură ridicată particulele solidului (aflat în faza lichidă) se aranjează aleator. Pe măsură ce temperatura scade, particulele tind să ajungă în stări cu energie din ce în ce mai mică. Dacă temperatura scade suficient de lent, atunci pentru fiecare valoare a temperaturii, solidului îi este permis să atingă starea de așa-numită echilibru termic.

### Principiul de funcționare

Elementele principale ale algoritmului Simulated Annealing sunt următoarele:

1. O mulțime finită S
2. O funcție de cost reală J definită pe S. Fie ⸦ S mulțimea minimelor globale ale funcției J, presupus a fi o submulțime corespunzătore al lui S
3. Pentru fiecare i ϵ S, o mulțime S(i) ⸦ S – {i}, numit mulțimea vecinilor lui i
4. Pentru fiecare i, o colecție de coeficienții pozitivi , j ϵ S(i), astfel încât = 1. Se presupune că j ϵ S(i) dacă și numai dacă i ϵ S(j)
5. O funcție necrescătoare T: N → (0, ∞), denumit program de răcire. N este mulțimea numerelor întregi pozitive, iar T(t) este temperatura la timpul t.
6. O “stare” inițială x(0) ϵ S

Având in vedere elementele de mai sus, algoritmul SA constă dintr-un lanț Markov neomogen discret în timp x(t). Dacă starea curentă x(t) este egală cu i, se alege un vecin j al lui i la întâmplare. Probabilitatea pentru oricare j ϵ S(i) să fie selectat este egală cu . Odată ce j este ales, următoarea stare x(t + 1) este determinată astfel:

* Dacă J(j) ≤ J(i), atunci x(t + 1) = j
* Dacă J(j) > J(i), atunci x(t + 1) = j cu probabilitatea exp[- (J(j) – J(i)) / T(t)], altfel x(t + 1) = i

În teorie, P[x(t + 1) = j|x(t) = i]

= exp (1)

dacă j ≠ i, j ϵ S(i).

Dacă j ≠ i și j ∉ lui S(i), atunci P[x(t + 1) = j | x(t) = i] = 0.

Rațiunea din spatele algoritmului Simulated Annealing este cel mai bine înțeleasă prin luarea în considerare a unui lanț Markov neomogen (t), în care temperatura T(t) este menținută la o valoare constantă T. Să presupunem că lanțul Markov (t) este ireductibil și aperiodic și că = pentru fiecare i, j. Atunci (t) este un lanț Markov reversibil și distribuția sa invariantă de probabilitate este dată de către:

(i) = exp, i ϵ S (2)

Unde este o constantă de normalizare. Este evident că atunci când T ↓ 0, distribuția de probabilități este concentrată pe mulțimea a minimelor globale al lui J. Această ultimă proprietate rămâne validă daca condiția că = este îndeplinită. (Faigle și Kern, 1989).

Distribuția probabilităților (2), cunoscut sub numele de distribuția Gibbs, joacă un rol foarte important în mecanica statistică. De fapt, fizicienii statisticieni au fost interesați să genereze un element eșantion de S, desenat în funcție de distribuția de probabilități . Aceasta se realizează simulând lanțul Markov (t) până când ajunge la echilibru, metoda fiind cunoscută sub numele de algoritmul Metropolis (1953). În contextul optimizării, se poate genera un element optim de S cu o probalitate mare dacă se va produce un eșantion aleatoriu în funcție de distribuția , cu o temperatură (T) foarte mică. O dificultate în această abordare este că atunci când T este foarte mic, timpul necesar lanțului Markov pentru a ajunge la echilibru poate fi excesiv. Algoritmul Simulated Annealing încearcă să remedieze acest dezavantaj utilizând un “program de răcire” lentă T(t).

Procedeul Simulated Annealing poate fi privit ca un algoritm de căutare locală în care (spre deosebire de algoritmul de căutare locală determinist) există mișcări ocazionale “în sus” care conduc la o creștere a costului. Se dorește ca astfel de mișcări ascendente să ajute la evitarea minimelor locale.

### Analiza convergenței

În acest subcapitol se vor prezenta performanțele algoritmului Simulated Annealing. De acum înainte, se va presupune că pentru un anumit T (și prin urmare pentru toate temperaturile), lanțul Markov (t) este ireductibil și aperiodic. Se va spune că procedeul Simulated Annealing converge dacă P[x(t) ϵ ] = 1. Un volum echitabil de muncă a fost utilizată în găsirea condițiilor necesare și suficiente pentru convergență. Rezultatul principal, datorat lui Hajek, este prezentat în continuare, urmând câteva definiții.

**Teorema 1 (Hajek, 1988):** Spunem că starea i comunică cu la o înălțime h dacă există un drum în S (fiecare element al căii fiind un vecin al elementului precedent) care începe de la i și se termină la un element al lui și astfel încât cea mai mare valoare a lui J de-a lungul drumului este J(i) + h. Fie cel mai mic număr astel încât fiecare i ϵ S comunică cu la înălțimea . Atunci, procedeul SA converge dacă și numai dacă T(t) = 0 și

**. (3)**

Cel mai popular program de răcire (cel puțin în teorie) este sub forma:

T(t) = , (4)

unde d este o constantă pozitivă. Teorema 1 afirmă că SA converge dacă și numai dacă d ≥ .

Constanta este o măsură a dificultății pentru x(t) pentru a evita minimele locale și a trece de la o stare neoptimală la . Suntem înteresați în primul rând de problemele în care > 0, care vor fi asumate de acum încolo. Astfel de probleme au minimele locale care nu sunt optime. Câteva aspecte ale teoremei 1 sunt furnizate de următorul argument. Se ia în considerare un minim local al cărui “drum” este . Procedeul SA realizează un număr infinit de încercări de a trece de minimul local, iar probabilitatea de succes al fiecărei încercări este de ordinul**. Atunci condiția (3) argumentează (datorită lemei Borel-Cantelli) că un un număr infinit de astfel de încercări vor fi îndeplinite. Într-adevăr, teorema lui Hajek, se realizează prin estimarea statisticilor timpilor de ieșire din anumite vecinătăți ai minimelor locale.**

**Fie** (i, t) = P[x(t) = i]. Dacă T(t) scade foarte încet, ca în cazul (4), atunci comportamentul lui x(t) pe intervale de timp destul de lungi, este de așteptat ca diferența dintre and (i, t) să fie mică în orice moment. Într-adevăr, una dintre primele dovezi de convergență sa bazat pe această idee, desi rezultatele au fost mai puțin clare decât în Teorema 1.

Pentru a obține o mai mare intuiție cu privire la interpretarea Teoremei 1, vom continua legătura dintre SA și familia corespunzătoare de lanțuri omogene Markov. Pentru acest scop, se va considera programul de răcire T(t) = . În general, statisticile despre lanțul Markov x(t) din cadrul unui program variabil lent de răcire T(t) rămân aproape neschimbate dacă se utilizează un program de răcire aferent, în care temperatura este menținută constant pentru perioade destul de lungi de timp. În cazul nostru, programul T(t) = poate fi aproximat astfel. Fie = 1 și = + exp(kd). Atuni fie (t) = , pentru ≤ t < . Se consideră segmetnul kth [, ] al programului din bucăți constante (t). În acest caz se confruntă cu un lanț reversibil omogen (t), iar în continuare vom aprofunda cât de repede poate să ajungă în starea de echilibru.

Se dorește studiul convergenței lanțului (t). Valorile proprii ale matricei de probabilitate de tranziție sunt reale. Timpul de relaxare este determinată de a doua cea mai mare valoare proprie , existând estimări bune cel puțin în limita k → ∞ (exemplu: Chiang și Chow, 1988). În particular, dacă costul funcției J are global minim unic, timpul de relaxare este aproximat de exp(k). Destul de interesant este că este egală cu aceeași constanta definită în Teorema 1, deși acest lucru este departe de adevăr. Aceasta oferă o altă interpretare a condiției de convergență d ≥ pentru programul (t). Dacă d < , atunci la fiecare temperatură se folosește (t) pentru o fracție neglijabilă al timpului de relaxare, deși nu este suficientă pentru (i, t) de a se apropria de . Pe de altă parte, dacă d > , atunci intervalul [, ] corespunde timpilor de relaxare al lui (t) cu ajutorul exp[k( - d)], ce implică ca (i, ) să fie foarte aproape de (i), cu k → ∞.

Se poate urmări, de asemenea, această aproximare prin programe constante în mod direct, fără a introduce valori proprii de estimare. Ideea principală este că la o temperatura joasă, statisticile unui lanț omogen pot fi limitate cu exactitate prin vizionarea lui ca un lanț Markov perturbat și prin utilizarea limitelor de abatere destul de mari.

Convergența algoritmului Simulated Annealing (în sensul definit anterior) este o proprietate încurajatoare, dar nu este suficientă pentru a fi un algoritm potrivit. Este nevoie să se știe viteza de convergență. Se poate demonstra că pentru oricare program T(t) = și pentru fiecare t

P[x(t) ∉ | x(0) ≥ , (5)

unde A și a sunt constante pozitive ce depind de funcția J și structura vecinătății. Dacă se dorește ca x(t) să fie în afara lui cu o probabilitate mai mică decât ꞓ, avem nevoie ca t ≥ .

Pentru o perspectivă mai practică, în timp ce se execută algoritmul, trebuie ținută evidența celei mai bune stări i întâlnite până acum și costul J(i) asociat. Dacă algoritmul trebuie să fie executat de pași de timp, fără a se ține cont de valoarea P(x() ∉ ). Mai degrabă, interesul se axează pe probabilitatea ca nici o stare a lui să fie vizitată pe perioada de execuție a algoritmului. Având în vedere un program de răcire de forma T(t) = , unde d > , poate fi arătat că această probabilitate este cel puțin pentru câteva constante pozitive A și a. Ea dispare cu cât t → ∞. Pe de altă parte, dacă temperatura este fixată la oricare valoare pozitivă (sau chiar la înfinit, care corespunde unei plimbări aleatorii), probabilitatea ca nici o stare din să fie vizitată în unități de timp, este cel mult pentru constantele pozitive corespunzătoare B și b. Deci, de foarte multe ori, performanța unei plimbări aleatorii pare să fie mai bună decât garanțiile de performanță ale procedeului SA. Punctul cheie este că analizele de mai sus, bazate pe un timp asimptotic, implică constante extrem de mari și sunt în mare măsură irelevante. Într-adevăr, constantele din estimările de mai sus sunt adesea mult mai mari decât cardinalitatea spațiului S. În special, analiza de mai sus nu poate stabili că algoritmul SA este întotdeauna cea mai bună soluție pentru o căutare exhaustiv.

### Comportamentul în practică

În ciuda lipsei unei justificări teoretice riguroase a vitezei sale de convergență, cercetătorii au utilizat acest procedeu pe o scară largă în diferite aplicații în ultimul deceniu. Există numeroase programe de răcire ce influențează calitatea soluțiilor obținute. În van Laarhoven și Aarts (1987), autorii compară trei programe diferite de răcire pentru problema de partiționare a graficelor și au observat că particularitatea soluției găsite de diferitele programe de răcire pot să difere cu până la 10 procente. O altă observație este că timpul de execuție poate fi exagerat de mare pentru anumite probleme.

Bohachevsky, Johnson și Stein (1986) au propus o procedură SA “generalizată” pentru probleme de optimizare continuă și au aplicat metoda lor la o problemă optimă de proiectare. Mulți cercetători au considerat Simulated Annealing ca un instrument pentru dezvoltarea unor modele experimentale optime. Exemple recente includ Currin și colaboratorii săi (1991), Meyer și Nachtsheim (1988) și Sacks și Schiller (1988). Variante ale algoritmului bazate pe ideile lui Bayesian au fost propuse de Laud, Berliner și Goel (1989), dar și de van Laarhoven și colaboratorii săi (1989).

În ansamblu, SA este un algoritm probabilistic de aproximare general aplicabil și ușor de implementat, fiind capabil să producă soluții bune pentru o problemă de optimizare, chiar dacă nu se întelege bine structura problemei. Cu toate acestea, se consideră că sunt necesare mai multe cercetări, atât teoretice, cât și experimentale, pentru a evalua în continuare potențialul metodei.

### Aplicații ale procedeului Simulated Annealing

* Pentru rețele neuronale:

1. Minimizarea funcției de energie în cazul învățării supervizate la rețelele feedforward. În acest caz sunt utilizați pentru minimizarea unor funcții cu multe minime locale și sunt definite pe un domeniu continuu.
2. Determinarea stării de energie minimă la rețelele recurente stohastice (modelul Hopfield pentru rezolvarea problemelor de optimizare sau mașina Boltzmann utilizată în rezolvarea problemelor de asociere). În acest caz sunt utilizați pentru determinarea unei configurații de cost minim dintr-un spațiu discret.

* Alte aplicații:

1. Prelucrarea imaginilor (eliminarea zgomotului și segmentare).
2. Rezolvarea problemelor de rutare.
3. Modelarea structurilor amorfe.
4. Analiza datelor obținute prin investigații bazate pe difracție cu raze X sau rezonanța magnetică nucleară.

## Mediul de dezvoltare Visual Studio

Microsoft Visual Studio este un mediu de dezvoltare integrat (IDE) de la Microsoft. Este folosit pentru a dezvolta programe de calculator, precum și site-uri web, aplicații web, servicii web și aplicații mobile. Visual Studio utilizează platforme de dezvoltare software Microsoft: Windows API, Windows Forms, Windows Presentation Foundation, Windows Store și Microsoft Silverlight. Poate produe atât cod nativ, cât și cod gestionat.

Visual Studio include un editor de cod care suportă IntelliSense (componenata de completare a codului), precum și refactorizarea codului. Debuggerul integrat funcționează atât ca un debugger la nivel de sursă, cât și ca un depanator la nivel de mașină. Alte instrumente încorporate includ un profil de cod, designer de formulare pentru construirea aplicațiilor GUI, designer web, designer de clasă și designer de schemă de baze de date. Acesta acceptă plugin-uri care îmbunătățesc funcționalitatea la aproape toate nivelele, inclusiv adăugarea de suport pentru sistemele de control al sursei (precum Subversion și Git) și adăugarea de noi seturi de instrumente, cum ar fi editorii și designerii vizuali pentru domeniul specific al limbajelor sau alte instrumente ce ajută la procesul de dezvoltare software (cum ar fi clientu Team Foundation Server: Team Explorer).

Acest IDE suportă 36 de limbaje de programare diferite și permite editorului de cod și depanatorului să suporte (în grade diferite) aproape orice limbaj de programare, cu condiția să existe un serviciu specific limbajului. Limbajele de programare încorporate sunt: C, C++, C++/CLI, Visual Basic .NET, C#, F#, JavaScript, TypeScript, XML, XSLT, HTML și CSS. Oferă suport și pentru alte limbaje precum Phyton, Ruby, Node.js. Pentru limbajul de programare Java, Microsoft nu mai oferă suport.

Ediția cea mai de bază este ediția comunitară, fiind disponibilă în mod gratuit. Sloganul pentru versiunea Visual Studio Community este “IDE gratuit și complet, pentru studenți, dezvoltatori open-source și dezvoltatori individuali”.

În acest moment, Microsoft oferă suport pentru versiunea Visual Studio 2019.



Figură 7. Logo Microsoft Visual Studio

## Controlul versiunilor de dezvoltare software GitHub/Git

### GitHub

Github este o companie americană care oferă hosting pentru controlul versiunilor de dezvoltare software folosind Git. Este o filială a companiei Microsoft, care a achiziționat compania în 2018 pentru 7,5 miliarde de dolari. GitHub oferă toate funcționalitățile distribuite de control al versiunilor și managementul codului sursă (MCS) ale Git-ului, precum și adăugarea propriilor carateristici. În plus, oferă controlul accesului și mai multe funcții de colaborare, cum ar fi urmărirea erorilor, solicitările de caracteristici, gestionarea sarcinilor și documentația pentru fiecare proiect.

Această companie oferă planuri pentru conturile gratuite, profesionale și de întreprindere. Conturile gratuite GitHub sunt utilizate în mod obișnuit pentru a găzdui proiecte open-source. Începând cu Ianuarie 2019, GitHub oferă depozite private nelimitate pentru toate planurile, inclusiv pentru conturile gratuite.



Figură 8. Logo GitHub

### Git

Git este un sistem distribuit de control al versiunii pentru urmărirea schimbărilor în codul sursă în timpul dezvoltării software-ului. Acesta este conceput pentru coordonarea muncii între programatori, dar poate fi folosit și pentru urmărirea modificărilor în orice set de fișiere. Obiectivele sale includ viteza, integritatea datelor și susținerea fluxurilor de lucru distribuite neliniar.

A fost creat de Linus Torvalds în 2005 pentru dezvoltarea kernelelului Linux, împreună cu alți dezvoltatori de kerneluri. Ca și în cazul celorlalte sisteme distribuite pentru controlul versiunii și spre deosebire de majoritatea sistemelor client-server, fiecare director Git de pe fiecare calculator este un depozit complet cu o istorie completă și abilități complete de urmărire a versiunilor, fara a se ține cont de accesul la rețea sau de un server central.



Figură 9. Logo Git

# Implementarea soluției adoptate

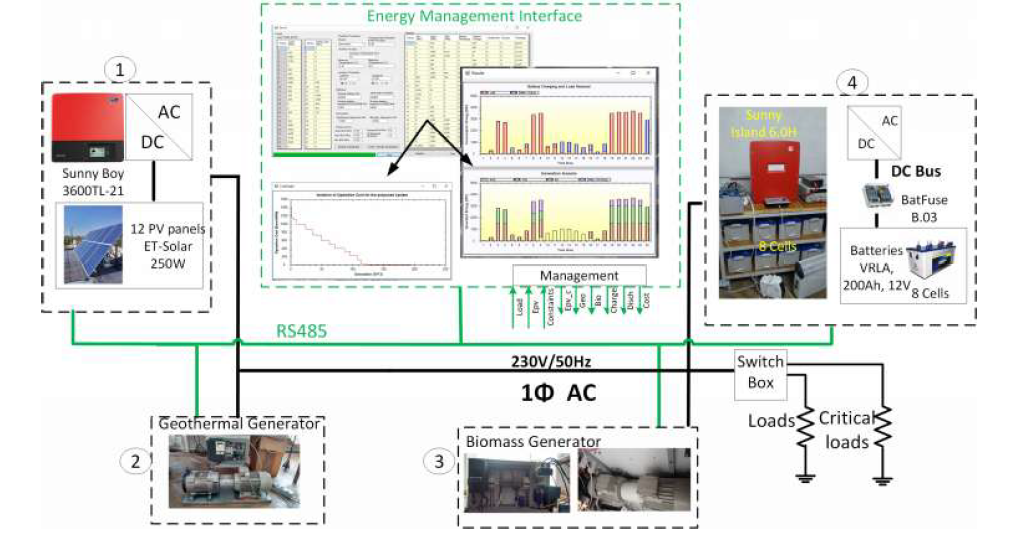
În acest capitol se prezintă soluția adoptată pentru întocmirea lucrării de diplomă. Capitolul se împarte în subcapitole pentru a se distinge mult mai bine toate aspectele abordate.

## Descrierea microrețelei

Diagrama schematică a microrețelei propuse cu 230V/50Hz este prezentată în Figura 10 [10]. Caracteristicile sistemului de stocare și a celor trei surse de energie distribuită utilizate sunt următoarele:

1. Panouri fotovoltaice de 12 – 250 Wp, fabricate de către ET Solar (ET-P660250WW), invertorul SMA Sunny-Boy 3600, o unitate ce umrărește transformarea razelor de soare în energie electrică. Modulele fotovoltaice sunt conectate în serie.
2. Un generator geotermal cu o putere maximă de ieșire de 3 kW.
3. Un generator de biomasă cu o putere maximă de ieșire de 3 kW.
4. Unități de stocare: 8 – 250 Ah, 12 V baterii VRLA, invertorul Sunny Island 6.0H. o unitate bidirecțională de formare a rețelei (setează voltajul și frecvența rețelei, fiind folosită pentru încărcarea și descărcarea bateriilor).

Comunicarea dintre componentele prezentate mai sus și PC este realizată cu ajutorul protocolului RS-485.



Figură 10. Microrețeaua propusă

## Formularea matematică

Scopul programării zilnice este de a obține costul minim de operare al microrețelei, asigurându-se în același timp că cerințele energetice sunt îndeplinite chiar și în momentele critice. Această problemă de optimizare poate fi reprezantată printr-o funcție obiectiv și mai multe constrângeri.

Funcția obiectiv al sistemului propus este reprezentată prin ecuația (6). Ea depinde de costul de operare al fiecărei unități conectate.

= (6)

Unde , și reprezintă energia produsă de soare, geotermă și biomasă, iar este energia încărcată / descărcată pentru baterii. reprezintă surplusul de energie produs. Acest exces de energie nu este stocată în baterii, dar nici nu este utilizată. , , , , și sunt coeficienții de cost. În Tabelul 1 sunt definiți coeficienții de cost pentru fiecare energie utilizată.

Tabel 1. Valoarea coeficienților de cost

|  |  |
| --- | --- |
| **Coeficientul de cost** | **€ / kWh** |
|  | 0.05 |
|  | 0.25 |
|  | 0.3 |
|  | 0.55 |
|  | 1.5 |

Mai mult decât atât, problema de optimizare prezentată este supusă la șapte constrângeri liniare. Acestea trebuie să fie validate în fiecare oră.

* Balanța energetică a microrețelei în intervale:

+ + + + - = 0 (7)

* Constrângerile panourilor fotovoltaice, energiei geotermale și de biomasă:

0 ≤ ≤ (8)

0 ≤ ≤ (9)

0 ≤ ≤ (10)

* Valorile minime și maxime de încărcare / descărcare:

-3500 Wh ≤ ≤ 1000 Wh (11)

* Limitele sistemului :

= - (12)

0 ≤ ≤ (13)

= 24000 Wh (14)

* Starea inițială de încărcare:

= 20 kWh (15)

Satisfacerea constrângerilor asigură faptul că cererea de sarcină este acoperită în fiecare etapă în timp ce se utilizează elementele microrețelei în parametrii specificați. Scopul probelemei de optimizare este de a minimiza funcția obiectiv, obținând astfel costul minim de operare al microrețelei.

Pentru microrețeaua propusă și pentru funcționarea ei sunt utilizate următoarele variabile de decizie: , , , și . Energia generată de fiecare sursă de energie regenerabilă (SER) , precum și energia în exces se calculează la fiecare oră. reprezintă energia electrică cerută la fiecare oră de către consumator. Prin utilizarea algoritmului pentru programarea zilnică și luând în considerare intervalul orar de timp, obținem 120 de variabile de decizie. Aceasta reprezintă dimensiunea problemei.

Limitele maxime și minime ale fiecărei variabile de decizie sunt stabilite utilizând ecuațiile de constrângere ale panourilor fotovoltaice, ale generatorului geotermal și de biomasă, precum și ale bateriilor (8) – (11). Panourile fotovoltaice sunt diferite de celelalte SER-uri în ceea ce privește domeniul lor. Toate SER-urile se caracterizează printr-o limită inferioară constantă (0 Wh), dar limitele superioare ale panourilor fotovoltaice, , depind de amplasarea microrețelei, precum și de condițiile meteorologice. Limitele stabilite în algoritm se bazează pe datele meteorologice statistice pentru locația dată a microrețelei. Generatoarele geotermale și de biomasă au o limită superioară constantă pentru generarea de energie. Limitele de încărcare și descărcare ale bateriilor sunt, de asemenea, constante.

Ecuația (6) oferă funcția obiectiv. Generarea de soluții inițiale, dar și noi, este supusă unor constrângeri de egalitate și de inegalitate. Acestea sunt reprezentate de ecuațiile (7), (15) și (13).

În Tabelul 2 și Tabelu 3 sunt prezentate limitele superioare ale panourilor fotovoltaice pentru fiecare lună din an, respectiv pentru fiecare oră. Valorile energiei electrice ce sunt cerute la fiecare oră de către consumator sunt definite în Tabelul 4.

Tabel 2. Limitele superioare ale panourilor fotovoltaice

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Ora** | **Ianuarie** | **Februarie** | **Martie** | **Aprile** | **Mai** | **Iunie** |
| **0** | 0 | 0 | 0 | 0 | 0 | 0 |
| **1** | 0 | 0 | 0 | 0 | 0 | 0 |
| **2** | 0 | 0 | 0 | 0 | 0 | 0 |
| **3** | 0 | 0 | 0 | 0 | 0 | 0 |
| **4** | 0 | 0 | 0 | 0 | 0 | 0 |
| **5** | 0 | 0 | 0 | 0 | 0 | 0 |
| **6** | 0 | 0 | 0 | 359 | 866 | 1080 |
| **7** | 0 | 0 | 366 | 1000 | 1503 | 1660 |
| **8** | 257 | 523 | 1033 | 1675 | 2128 | 2214 |
| **9** | 720 | 1141 | 1695 | 2290 | 2667 | 2685 |
| **10** | 1114 | 1668 | 2237 | 2760 | 3057 | 3022 |
| **11** | 1347 | 2001 | 2569 | 3023 | 3258 | 3195 |
| **12** | 1370 | 2074 | 2639 | 3047 | 3248 | 3186 |
| **13** | 1176 | 1876 | 2437 | 2829 | 3030 | 2996 |
| **14** | 811 | 1445 | 1997 | 2399 | 2628 | 2648 |
| **15** | 357 | 865 | 1390 | 1815 | 2090 | 2177 |
| **16** | 0 | 256 | 719 | 1157 | 1477 | 1633 |
| **17** | 0 | 0 | 97 | 513 | 858 | 1072 |
| **18** | 0 | 0 | 0 | 0 | 297 | 544 |
| **19** | 0 | 0 | 0 | 0 | 0 | 93 |
| **20** | 0 | 0 | 0 | 0 | 0 | 0 |
| **21** | 0 | 0 | 0 | 0 | 0 | 0 |
| **22** | 0 | 0 | 0 | 0 | 0 | 0 |
| **23** | 0 | 0 | 0 | 0 | 0 | 0 |

Tabel 3. Limitele superioare ale panourilor fotovoltaice (continuare)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Ora** | **Iulie** | **August** | **Septembrie** | **Octombrie** | **Noiembrie** | **Decembrie** |
| **0** | 0 | 0 | 0 | 0 | 0 | 0 |
| **1** | 0 | 0 | 0 | 0 | 0 | 0 |
| **2** | 0 | 0 | 0 | 0 | 0 | 0 |
| **3** | 0 | 0 | 0 | 0 | 0 | 0 |
| **4** | 0 | 0 | 0 | 0 | 0 | 0 |
| **5** | 0 | 0 | 0 | 0 | 0 | 0 |
| **6** | 1035 | 785 | 422 | 111 | 0 | 0 |
| **7** | 1593 | 1337 | 924 | 524 | 181 | 0 |
| **8** | 2132 | 1885 | 1442 | 972 | 538 | 300 |
| **9** | 2595 | 2366 | 1906 | 1383 | 878 | 643 |
| **10** | 2933 | 2727 | 2256 | 1688 | 1135 | 917 |
| **11** | 3116 | 2928 | 2446 | 1840 | 1257 | 1059 |
| **12** | 3125 | 2949 | 2452 | 1815 | 1221 | 1038 |
| **13** | 2961 | 2789 | 2273 | 1617 | 1033 | 859 |
| **14** | 2640 | 2465 | 1934 | 1279 | 733 | 562 |
| **15** | 2198 | 2016 | 1481 | 857 | 380 | 217 |
| **16** | 1679 | 1491 | 974 | 418 | 44 | 0 |
| **17** | 1136 | 946 | 478 | 29 | 0 | 0 |
| **18** | 618 | 438 | 48 | 0 | 0 | 0 |
| **19** | 169 | 10 | 0 | 0 | 0 | 0 |
| **20** | 0 | 0 | 0 | 0 | 0 | 0 |
| **21** | 0 | 0 | 0 | 0 | 0 | 0 |
| **22** | 0 | 0 | 0 | 0 | 0 | 0 |
| **23** | 0 | 0 | 0 | 0 | 0 | 0 |

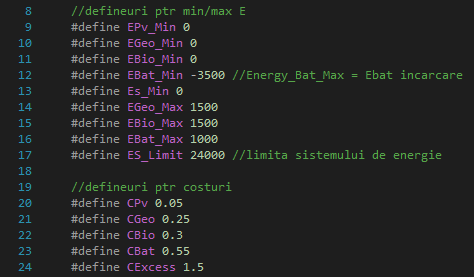
Tabel 4. Valorile energiei electrice ce sunt cerute la fiecare oră de către consumator

|  |  |
| --- | --- |
| **Ora** |  |
| 0 | 0 |
| 1 | 250 |
| 2 | 2800 |
| 3 | 2700 |
| 4 | 0 |
| 5 | 300 |
| 6 | 600 |
| 7 | 3400 |
| 8 | 3500 |
| 9 | 600 |
| 10 | 600 |
| 11 | 100 |
| 12 | 200 |
| 13 | 0 |
| 14 | 0 |
| 15 | 300 |
| 16 | 0 |
| 17 | 200 |
| 18 | 3500 |
| 19 | 3600 |
| 20 | 3600 |
| 21 | 3700 |
| 22 | 3500 |
| 23 | 0 |

## Implementarea în limbajul C

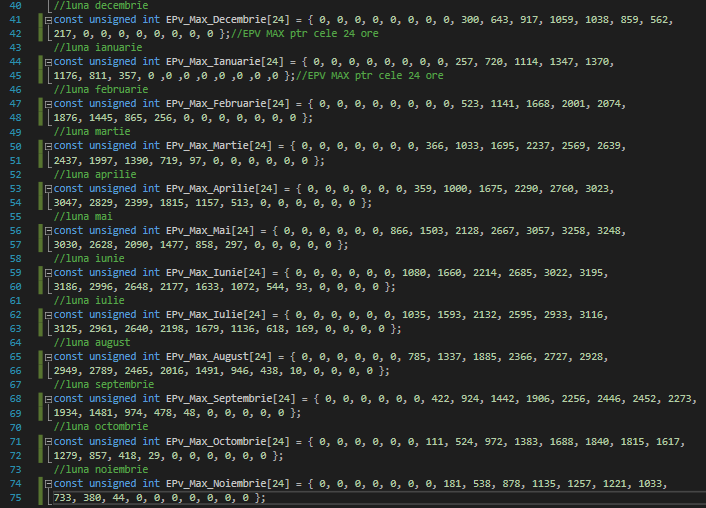
Algoritmul Simulated Annealing, procedeul folosit pentru realizarea proiectului de diplomă, a fost dezvoltat, implementat și testat în limbajul de programare C, cu ajutorul IDE-ului Microsoft Visual Studio.

În Figura 11 s-a definit în aplicație energiile și costurile necesare pentru rezolvarea problemei planificării zilnice. Aceste valori sunt preluate din Tabelul 1 și din ecuațiile (8), (9), (10), (11) și (14).



Figură 11. Definirea în program a energiilor și costurilor

În Figura 12 s-a definit în program limitele superioare ale panourilor fotovoltaice pentru fiecare lună din an. Valorile sunt preluate din Tabelul 2 și Tabelul 3.



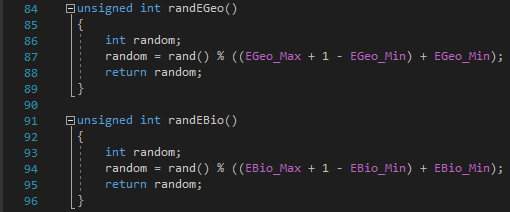
Figură 12. Limitele superioare ale panourilor fotovoltaice

În Figura 13 s-a definit energia electrică cerută la fiecare oră de către consumator.



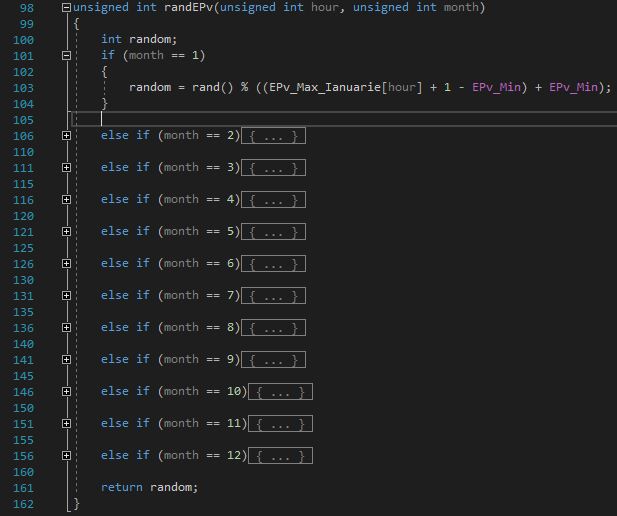
Figură 13. Energia electrică cerută la fiecare oră de către consumator

În Figura 14 s-a realizat funcțiile ce realizează o valoare aleatoarie pentru energia geotermală și de biomasă, respectând constrângerile (9) și (10).



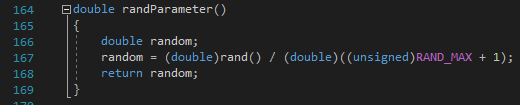
Figură 14. Funcțiile ce realizează o valoare aleatorie, respectând constrângerile pentru energia geotermală și de biomasă

În Figura 15 s-a realizat o funcție ce produce o valoare aleatorie pentru enegia fotovoltaică, respectând constrângerea (8) și limitele superioare ale panourilor fotovoltaice în funcție de lună.



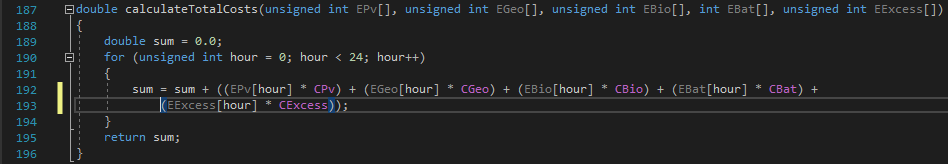
Figură 15. Funcție ce realizează o valoare aleatorie, respectând constrângerea pentru energia fotovoltaică

În Figura 16 s-a realizat o funcție ce produce o valoare aleatorie în intervalul [0,1], utilizată de către algoritmul Simulated Annealing pentru rezolvarea problemei planificării zilnice.



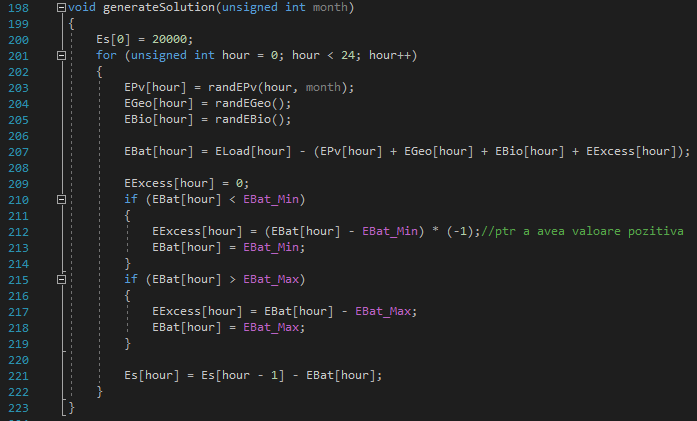
Figură 16. Funcție ce produce o valoare aleatorie utilizată în procedeul Simulated Annealing

În Figura 17 s-a implementat ecuația (6). Ecuație ce realiează costul total al unei soluții.



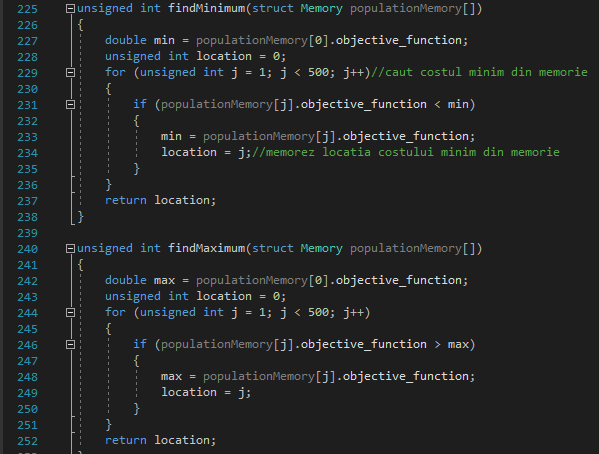
Figură 17. Implementarea ecuației (6)

În Figura 18 s-a realizat o funcție ce generează o soluție nouă respectând ecuațiile (7), (11), (12), (13) și (14).



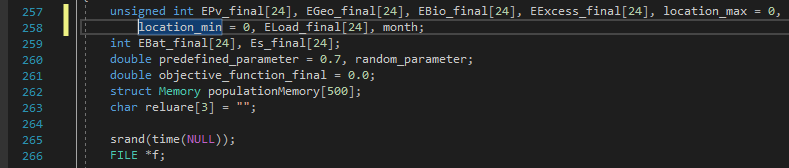
Figură 18. Funcție ce realizează o nouă soluție

În Figura 19 s-a implementat 2 funcții ce caută soluția cu cost minim sau maxim din populația de soluții (din memorie).



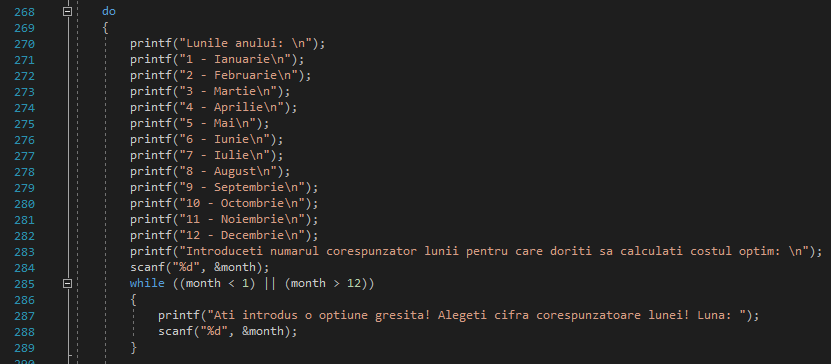
Figură 19. Funcții ce caută soluția cu cost minim/maxim din memorie

În Figura 20 sunt definite variabilele locale și inițializarea funcției predefinite rand() cu ajutorul clock-ului intern al PC-ului.



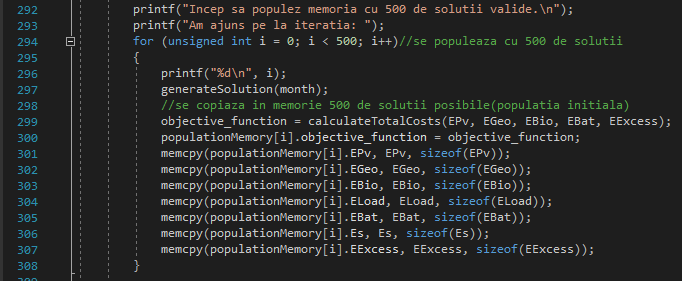
Figură 20. Variabile locale

În Figura 21 s-a realizat interacțiunea cu utilizatorul.



Figură 21. Interacțiunea cu utilizatorul

În Figura 22 s-a realizat popularea memoriei cu 500 de soluții valide.



Figură 22. Popularea memoriei cu 500 de soluții valide

În Figura 23, Figura 24, Figura 25 și Figura 26 s-a implementat primul caz folosit de către procedeul Simulated Annealing pentru rezolvarea problemei planificării zilnice. Pentru a se executa primul caz, parametrul “random\_parameter” din aplicație (fiind un parametru ce primește o valoare aleatorie din intervalul [0,1]) trebuie să fie mai mic sau cel mult egală cu parametrul predefinit (în aplicație parametrul predefinit este egală cu valoarea 0.7).

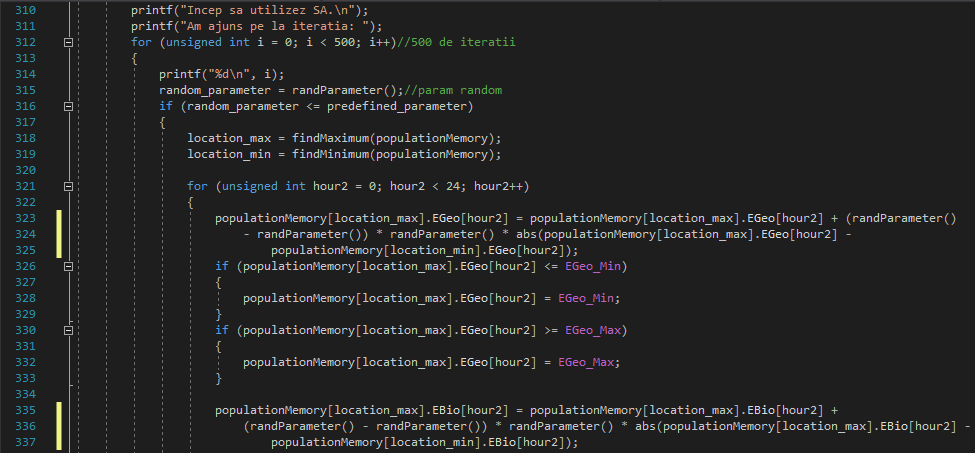
Primul caz presupune alterarea celei mai costisitoare soluții din populația inițială, pentru a forma o nouă soluție în memorie. Alterarea soluției se realizează cu ajutorul următoarei ecuații:

= + ( - ) \* \* abs(max -min) (16)

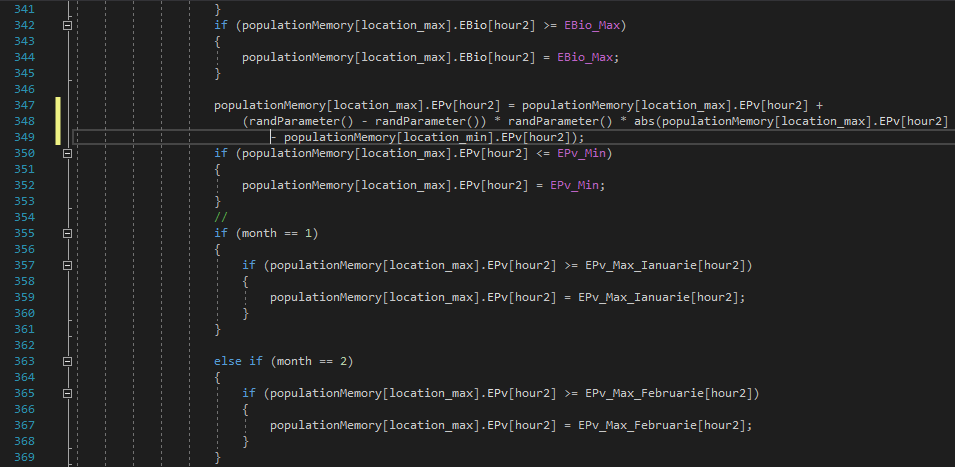
Unde:

* Random1, random2 și random3 sunt valori alteatorii din intervalul [0,1]
* reprezintă soluția ce este alterată
* Max reprezintă soluția cu cel mai mare cost din memorie
* Min reprezintă soluția u cel mai mic cost din memorie

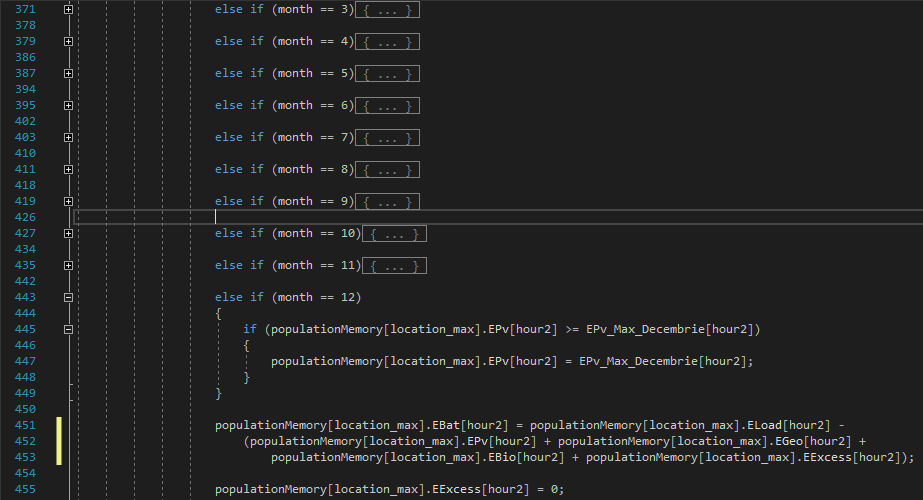
Variabilele de decizie EGeo (energia generată de către generatorul geotermal), EBio (energia generată de către generatorul de biomasă) și EPv (energia generată de către panourile fotovoltaice) sunt alterate folosind ecuația (16). Desigur că și celelalte variabile ce depind de EGeo, EBio și Epv sunt alterate și ele în urma acestui proces.



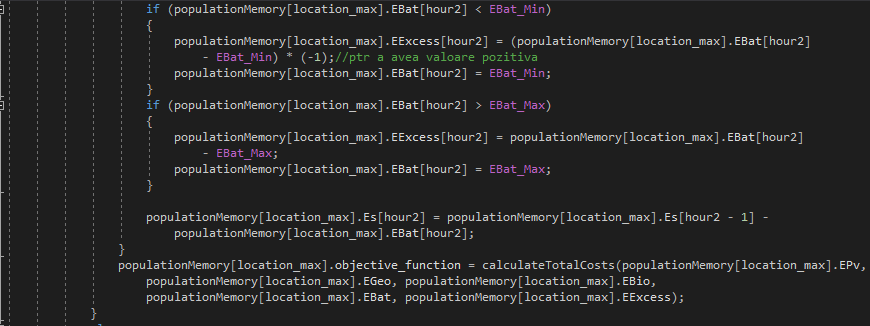
Figură 23. Primul caz al algoritmului Simulated Annealing (I)



Figură 24 Primul caz al algoritmului Simulated Annealing (II)



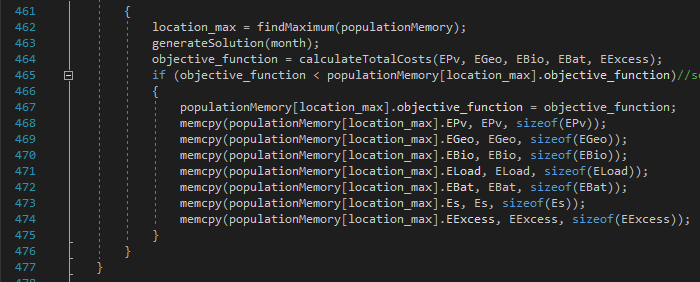
Figură 25. Primul caz al algoritmului Simulated Annealing (III)



Figură 26. Primul caz al algoritmului Simulated Annealing (IV)

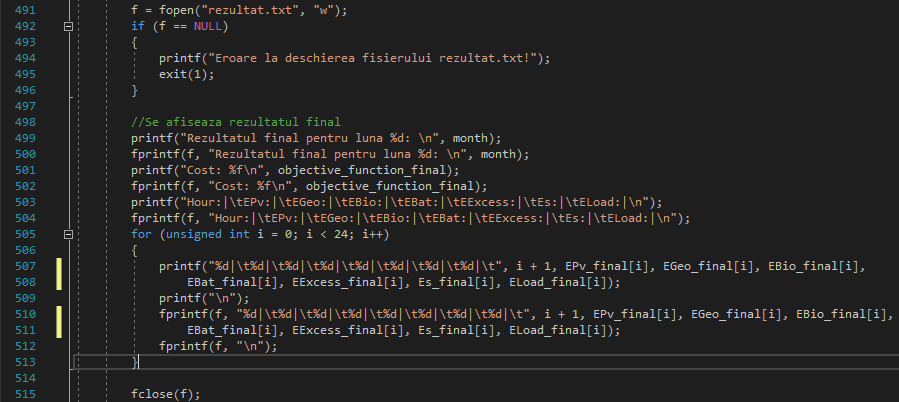
În Figura 27 s-a implementat al doilea caz folosit de către procedeul Simulated Annealing. Pentru a se executa al doilea caz, parametrul “random\_parameter” din program trebuie să fie strict mai mare de valoarea 0.7, această valoare fiind parametrul predefinit (“predefined\_parameter” din aplicație).

Al doilea caz constă în suprascrierea celei mai costisitoare soluții din memorie cu o nouă soluție generată conform funcției definite în Figura 18.

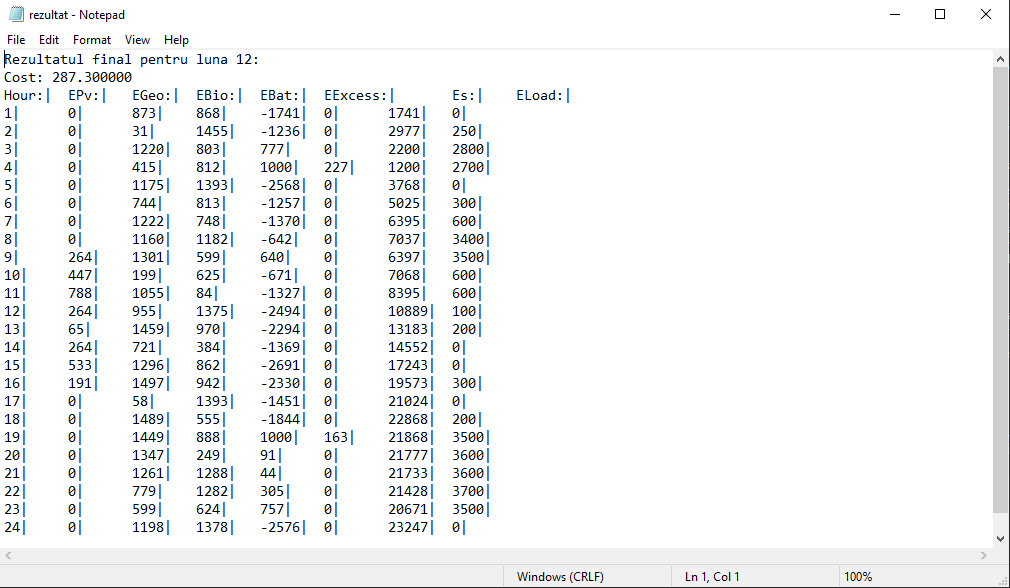


Figură 27. Al doilea caz al procedeului Simulated Annealing

În Figura 28 s-a implementat afișarea rezultatului final la consolă, dar și memorarea ei în fișierul “rezultat.txt”.

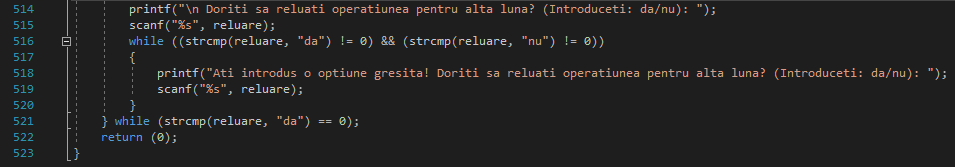


Figură 28. Afișarea rezultatului la consolă și memorarea în fișier



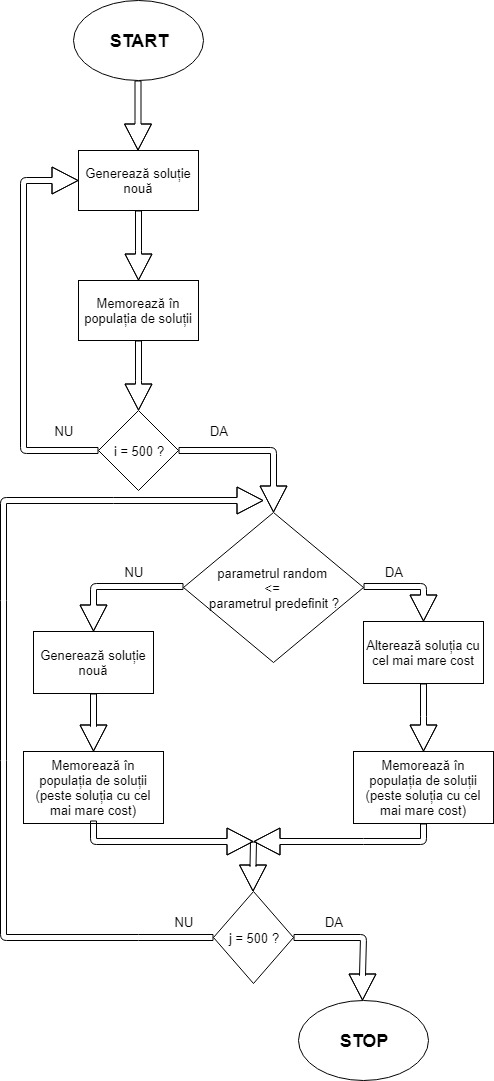
Figură 29. Exemplu de memorare a rezultatului în fișierul “rezultat.txt”.

În Figura 30 am implementat un nou caz de interacțiune cu utilizatorul. Acesta este întrebat de către aplicație dacă dorește să realizeze o nouă operațiune de calculare a costului minim pentru o altă lună a anului.



Figură 30. Interacțiunea cu utilizatorul (II)

## Organigrama algoritmului Simulated Annealing

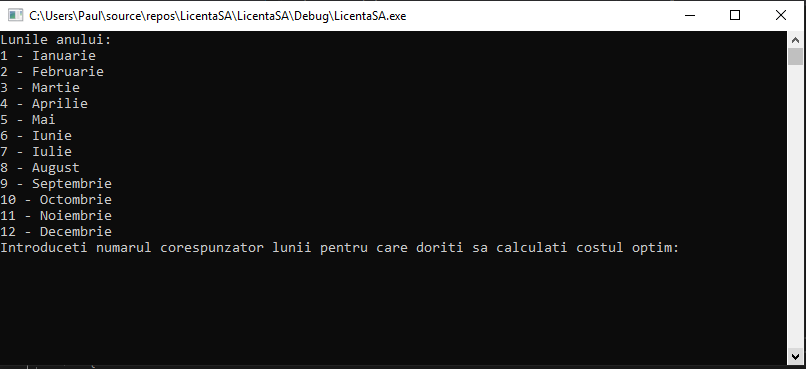


# Rezultate experimentale

În urma procesului de dezvoltare și testare a algoritmului Simulated Annealing, cu ajutorul IDE-ului Microsoft Visual Studio, s-au obținut rezultatele prezentate mai jos.

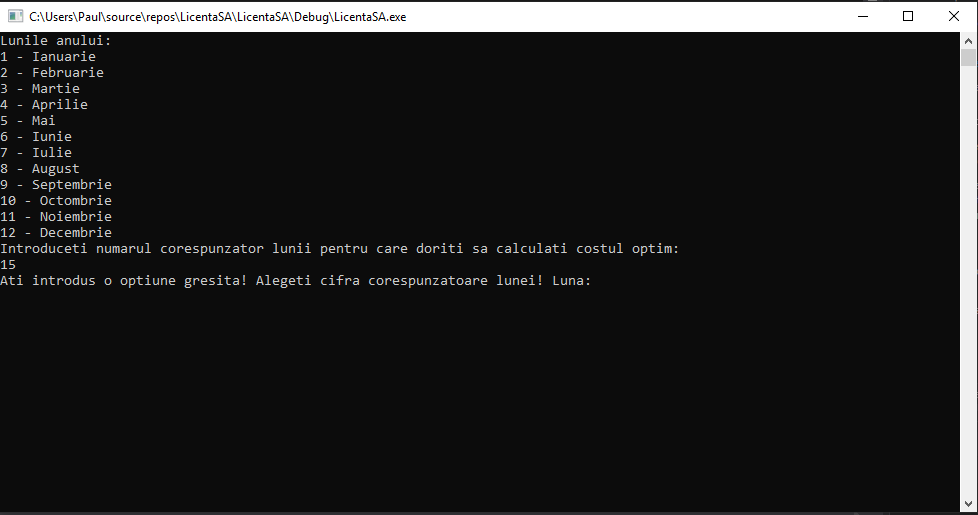
## Meniul pentru alegerea lunii

În Figura 12 este prezentat un meniu cu 12 opțiuni disponibile, reprezentând cele 12 luni ale anului. Utilizatorul poate selecta din lista respectivă luna pentru care algoritmul Simulated Annealing să realizeze costul optim de operare al microrețelei. Acest meniu a fost implementat pentru a avea o interacțiune mult mai prietenoasă cu utilizatorul.



Figură 31. Meniul pentru alegerea lunii

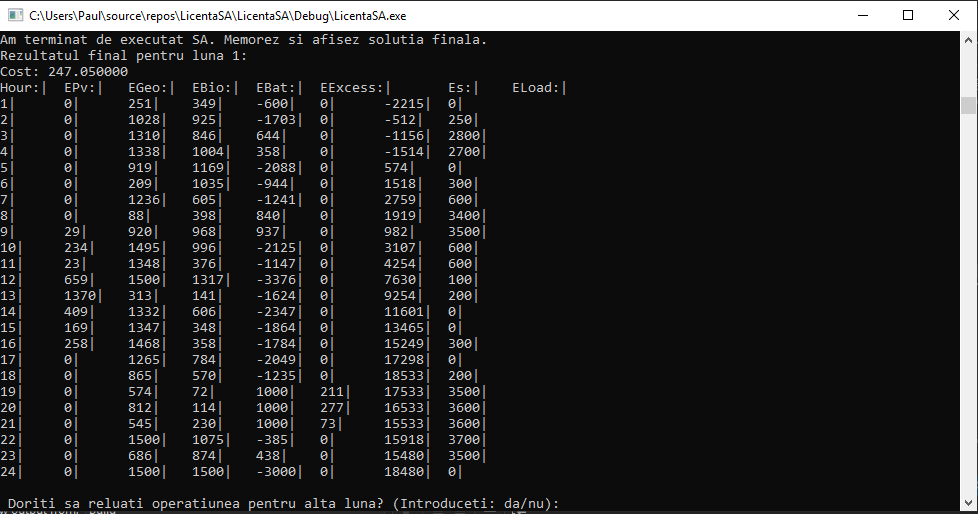
În cazul în care utilizatorul introduce o altă opțiune care nu este specificată în lista meniului, aplicația o să îl atenționeze că a introdus o opțiune greșită și așteaptă ca acesta să introducă o opțiune validă.



Figură 32. Cazul în care se introduce o opțiune inexistentă

## Luna Ianuarie

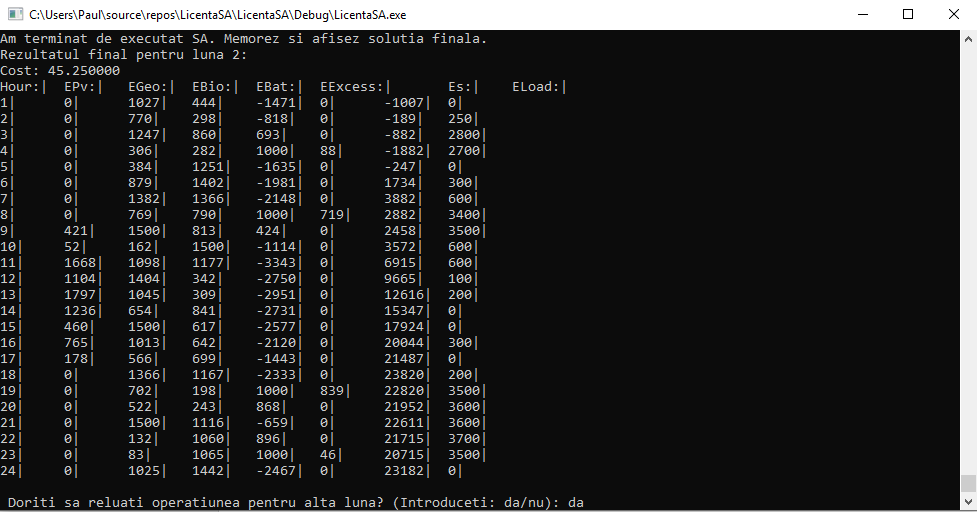
În Figura 14 se poate observa costul optim calculat de către procedeul Simulated Annealing pentru luna Ianuarie. Cost: 247.05.



Figură 33. Costul pentru luna Ianuarie

## Luna Februarie

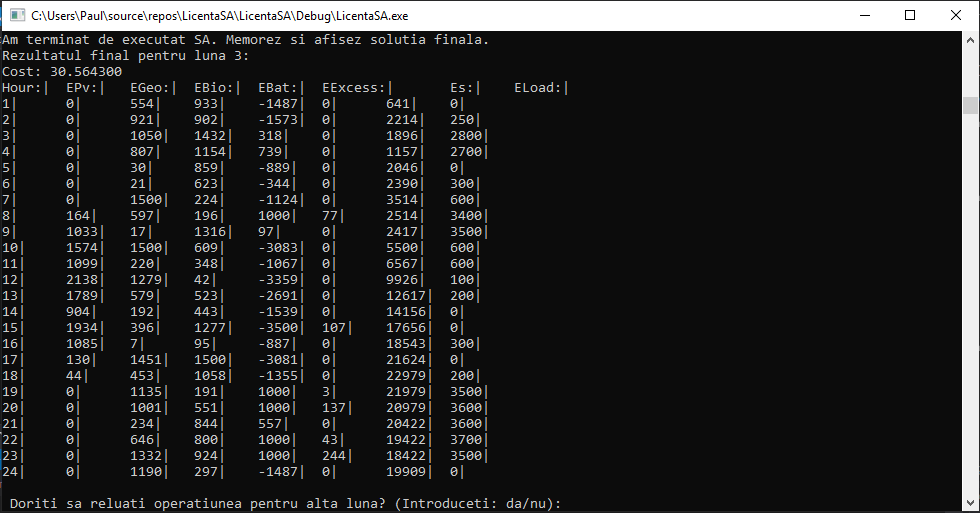
În figura de mai jos se poate observa costul pentru luna Februarie. Cost: 45.25



Figură 34. Costul pentru luna Februarie

## Luna Martie

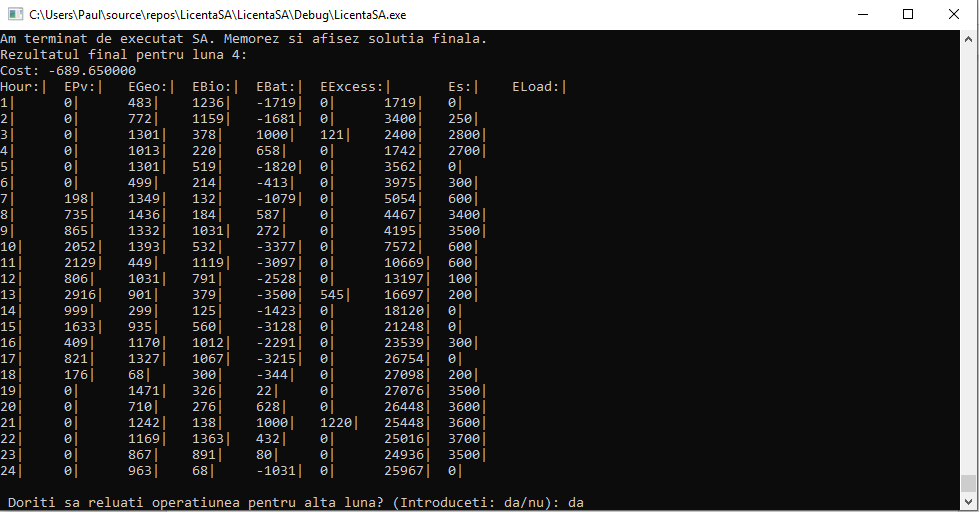
În figura de mai jos se poate observa costul pentru luna Martie. Cost: 30.5643



Figură 35. Costul pentru luna Martie

## Luna Aprilie

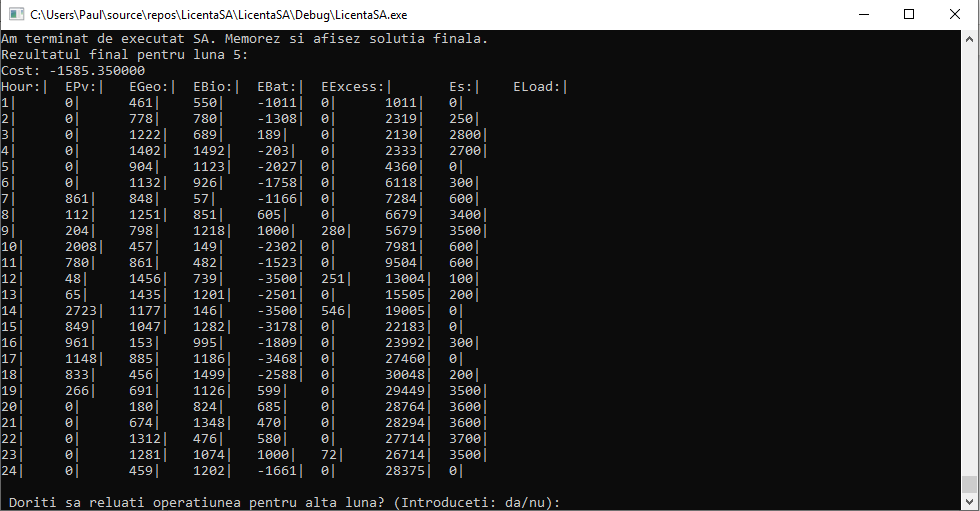
În figura de mai jos se poate observa costul pentru luna Aprilie.



Figură 36. Costul pentru luna Aprilie

## Luna Mai

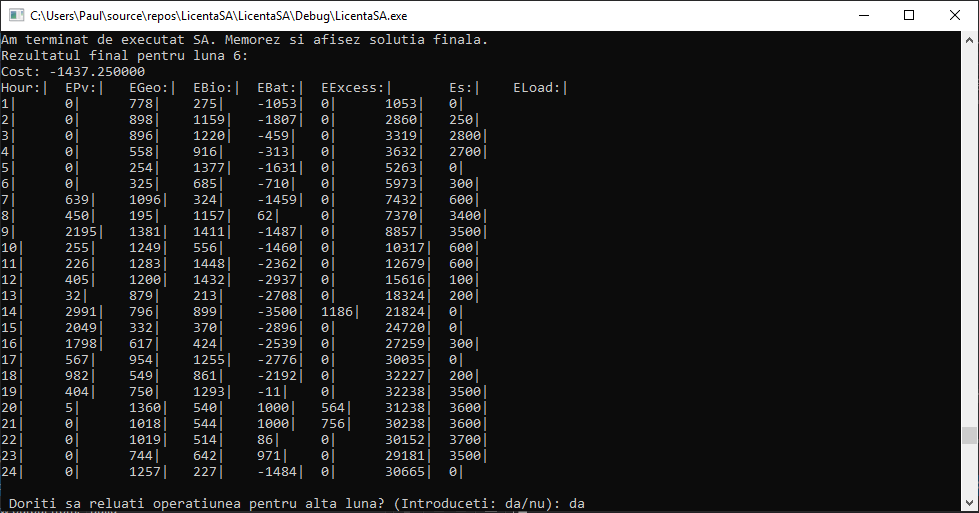
În figura de mai jos se poate observa costul pentru luna Mai.



Figură 37. Costul pentru luna Mai

## Luna Iunie

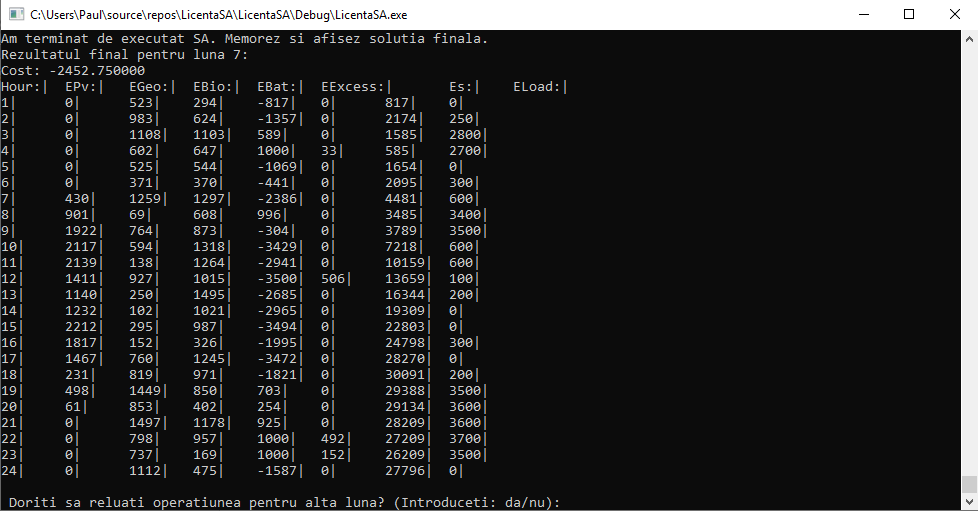
În figura de mai jos se poate observa costul pentru luna Iunie.



Figură 38. Costul pentru luna Iunie

## Luna Iulie

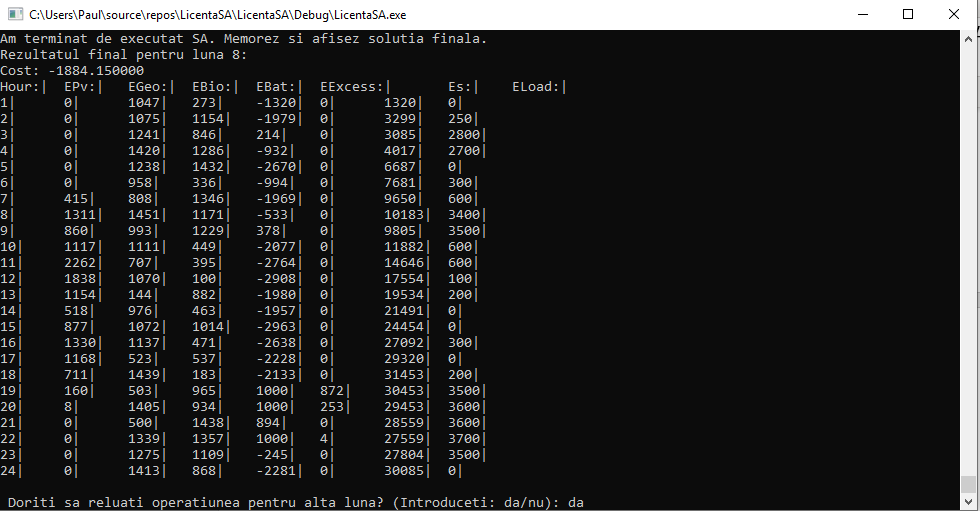
În figura de mai jos se poate observa costul pentru luna Iulie.



Figură 39. Costul pentru luna Iulie

## Luna August

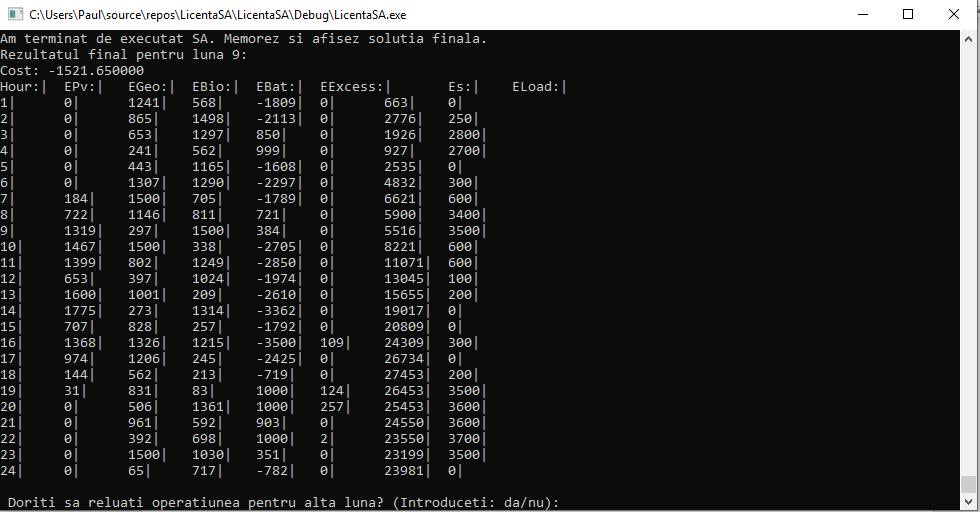
În figura de mai jos se poate observa costul pentru luna August.



Figură 40. Costul pentru luna August

## Luna Septembrie

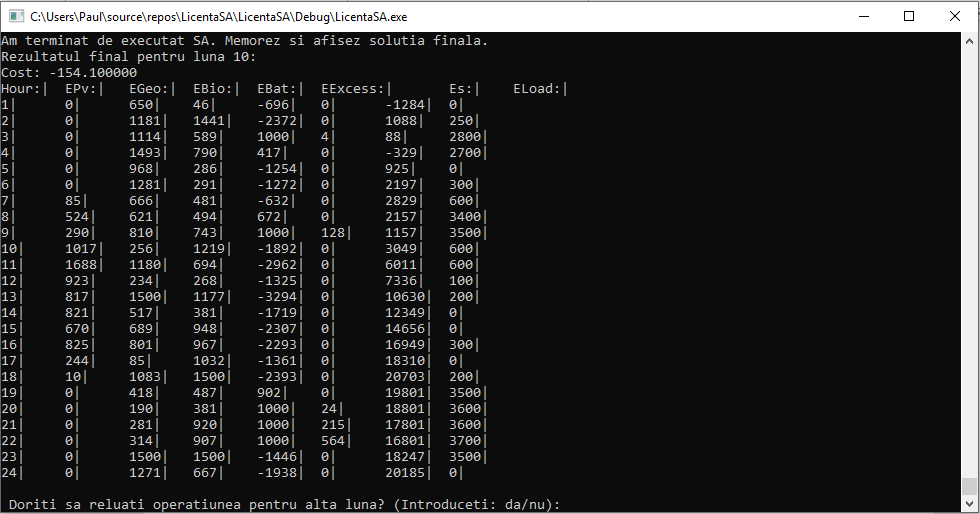
În figura de mai jos se poate observa costul pentru luna Septembrie.



Figură 41. Costul pentru luna Septembrie

## Luna Octombrie

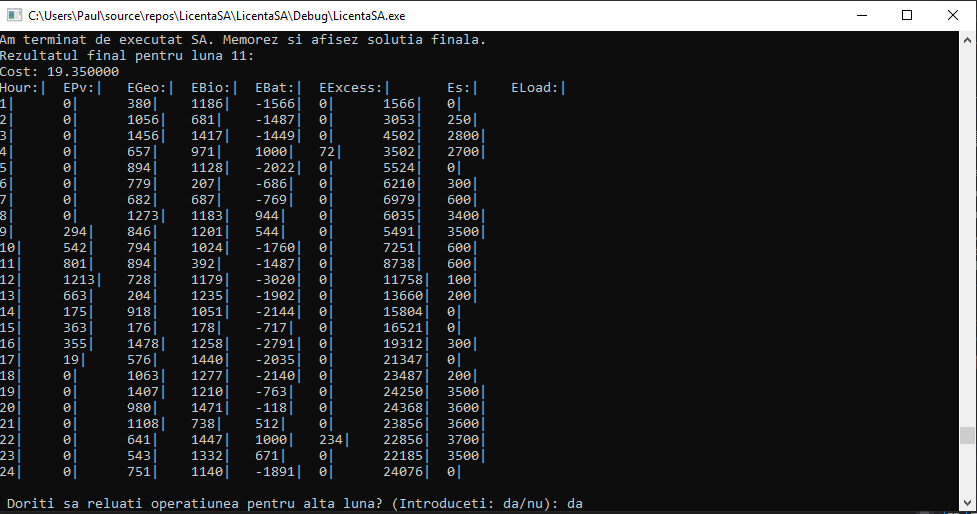
În figura de mai jos se poate observa costul pentru luna Octombrie.



Figură 42. Costul pentru luna Octombrie

## Luna Noiembrie

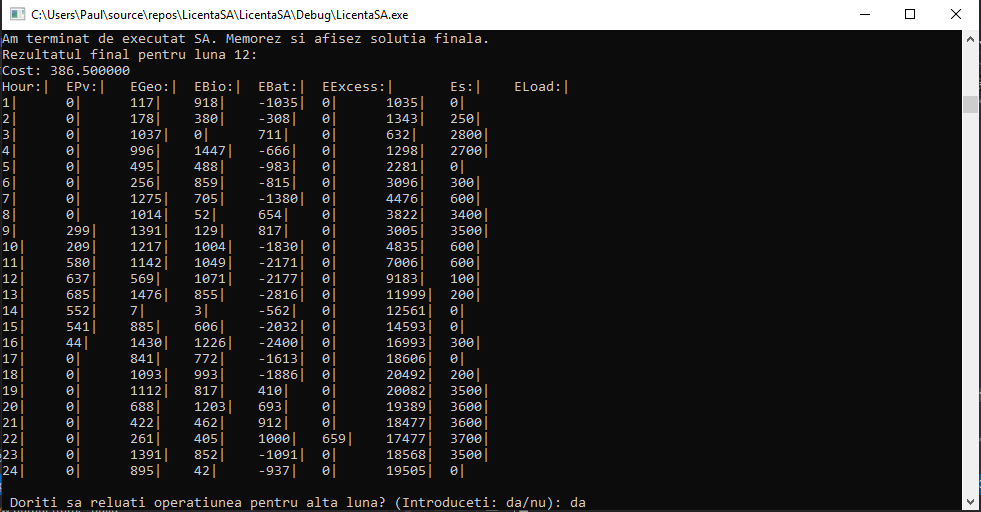
În figura de mai jos se poate observa costul pentru luna Noiembrie. Cost: 19.35



Figură 43. Costul pentru luna Noiembrie

## Luna Decembrie

În figura de mai jos se poate observa costul pentru luna Decembrie. Cost: 19.35



Figură 44. Costul pentru luna Decembrie

Bibliografie

1. Geem ZW, Kim JH, Loganathan GV “A new heuristic optimization algorithm: Harmony Search”, Simulation, vol. 76, pp. 60-68, February 2001
2. Lee KS, Geem ZW “A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice”, Computer Methods in Applied Mechanics and Engineering, vol. 194, pp. 3902-3933, September 2005
3. Kennedy, J.; Eberhart, R. (1995). “Particle Swarm Optimization”. Proceedings of IEEE International Conference on Neural Networks. IV. pp. 1942-1948
4. B. Borowska, “Nonlinear inertia weight in particle swarm optimization,” 2017 12th International Scientific and Technical Conference on Computer Sciences and Information Technologies (CSIT), Lviv, 2017, pp. 296-299.
5. www.smartgrids.eu
6. www.galvinelectricity.org
7. Comisia Europeană – “Vision and Strategy for Europe’s Electricity Networks of the Future”, Platforma Tehnologică Europeană SmartGrids, 2008
8. ERPI – “The integrated Energy and Communication Systems Arhitecture”, Vol. IV, Technical Analysis, Electric Power Research Institute, 2004
9. Simulated Annealing.pdf
10. ISSE2018\_E\_.pdf
11. Aldoileapdf.pdf
12. Neural Network Toolbox Help, Matlab R2010a.