



Sesiunea de Comunicări Științifice Studențești 2020

Rolul Centralelor Eoliene In Estimare Cererii De Energie Într-un Micro Grid

Semion PELESKOV
Ana-Maria-Ruxandra REȘCANU

2020

“Energia produsă de centralele eoliene este redirecționată prin rețele inteligente “Microgrid”, folosind un algoritm de redistribuire specific. Lucrarea de față realizează un studiu asupra estimării cererii de energie prin scenariul amplasării turbinelor eoliene în zona unui oraș dezvoltat, respectiv, într-o zonă rurală din România. Studiul s-a realizat mai întâi prin modelarea distribuției vitezei vântului pornind de la Repartiția Weibull, iar estimarea producției de energie s-a efectuat utilizând Ecuația Analitică a Producției de Energie Eoliană. Pentru estimarea parametrului de scală din Repartiția Weibull s-a folosit un program creat pe baza Gramaticilor Evolutive.”

“The energy produced by wind power plants is being redistributed by smart networks named “Micro Grid” by applying of a specific algorithm. This paper conducts a study on the estimation of energy demand through the scenario of the placement of wind turbines in the area of a developed city, respectively, in a countryside of Romania. The study was first performed by modeling the wind speed relation starting from the Weibull Distribution, then the energy production estimation was performed using the Analytical Equation of Wind Energy Production. Further, a program based on the concept of Grammatical Evolution was created in order to estimate the scale parameter in the Weibull Distribution.”

Coordonator Științific:
Lect. Dr. Mihai Rebenciuc

CUPRINS

1. INTRODUCERE	3
2. BIOMIMETISMUL ÎN PROGRAMARE	4
2.1 Programare Genetică	4
2.2 Gramatici Evolutive	5
3. REPARTIȚIA WEIBULL. MODELAREA VITEZEI VÂNTULUI	8
3.1 Formalism Weibull	8
4. STUDIU DE CAZ	9
5. CONCLUZII	13
6. REFERINȚE	14

INTRODUCERE

Rețelele inteligente (Micro Grid) reprezentate în fig. 1, sunt noduri ale unei rețele mai mari (Smart Grid). „Micro Grid” reprezintă un proiect propus în cadrul conceptului “Smart City”, care la rândul său sugerează schimbări de viziune în sfera organizării infrastructurii unei așezări umane. “Micro Grid” are la bază principii inteligente de producție, stocare și redistribuire a energiei. Ideea principală constă în faptul că energia din aceste rețele nu este pierdută niciodată, ci este redirecționată în nodurile unde cererea energiei este mai ridicată. În plus, “Micro Grid” are avantajul de a fi complet autonom, atât la oraș, cât și în zonele rurale, în cazul în care sunt avarii la Sistemul de Energie Național (SEN). [2][3]

Energia eoliană necesită anticiparea distribuției vitezei vântului pentru a estima valoarea producției sale. Valoarea estimării va fi raportată dispeceratului energetic, pentru a regla fluxul de energie provenit de la hidrocentrale, respectiv termocentrale. În cazul unui “Micro Grid”, pe baza acestor predicții se poate estima cererea de energie în funcție de zona în care el este amplasat, și de factorii climatici ai acestei zone.

Astfel, se identifică două probleme: modelarea distribuției vitezei vântului (WSD) și simularea estimării cantității de energie produsă. În lucrarea de față, distribuția vitezei vântului este modelată, prin două metode: Metoda Maximei Verosimilități asociată Repartiției lui Weibull, și prin aplicarea Gramaticilor Evolutive. Cantitatea de energie produsă este estimată prin Ecuația Analitică a Producției de Energie Eoliană.

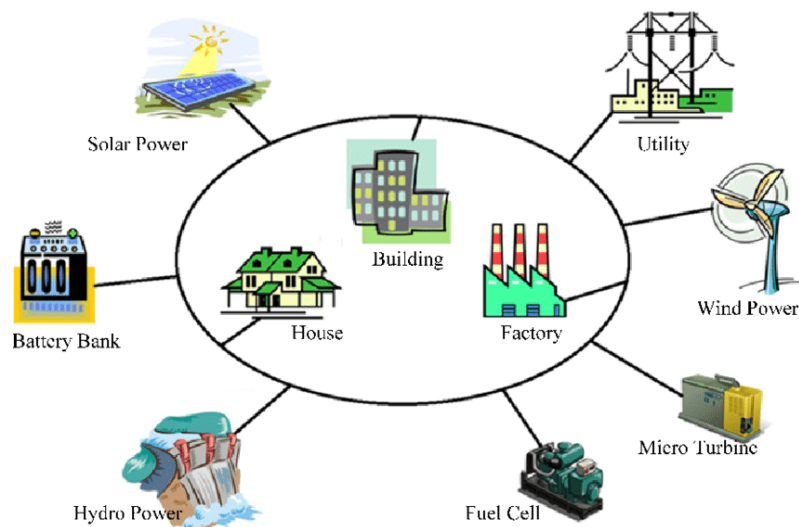


Fig. 1: Schema generală unui “Micro Grid”^[1]

BIOMIMETISMUL ÎN PROGRAMARE

Multe domenii ale vieții umane depind de tehnologii, la baza cărora stau principii ce au fost împrumutate din natură. BIA (Biologically Inspired Algorithms) sunt tehnologiile care stau la baza Programării Genetice (GP) și a Gramaticilor Evolutive (GE).^{[7][8]} Acești algoritmi se folosesc de terminologia și principiile enunțate de Charles-Robert Darwin în anul 1859 în lucrarea sa “On The Origin Of Species”.^[4]

Algoritmul Genetic (GA)^[7] reprezintă o tehnică de obținere a punctului de minim al unei funcții multidimensionale. În limbajul teoriei evoluționiste, rezultatul Algoritmului Genetic produce o populație care se potrivește cel mai bine unui mediu prezentat în problema. În fig. 2 este reprezentată legătura între formalisme de programare aplicate.

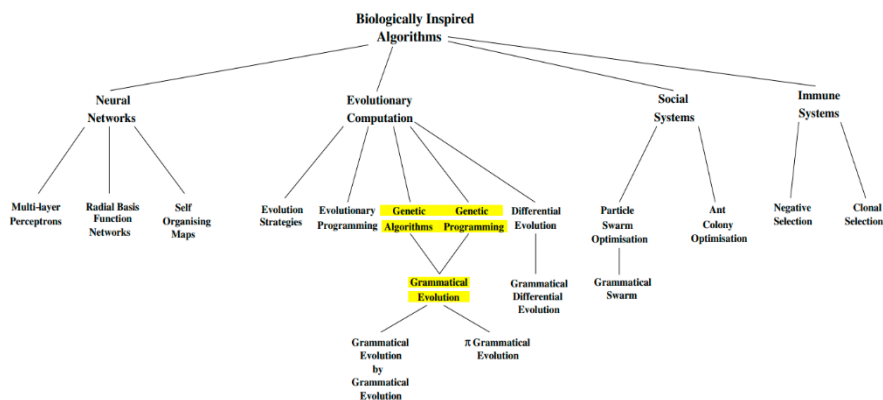


Fig. 2: Schema de tip arbore a principalelor BIA cu evidențiere de cele aplicate în lucrare^[7]

1. PROGRAMARE GENETICĂ

GP are două diferențe fundamentale față de GA. În loc să evolueze șiruri binare, care reprezintă o codificare indirectă a unei soluții potențiale, în evoluția GP căutarea este aplicată direct soluției, în acest caz, soluțiile fiind programe. În formele de GP promovate de John Koza, aceste programe au forma unei “S-expression”, care reprezintă o schemă de sintaxă. În fig. 3 este prezentat un exemplu de codificare al unui rând:^[7]

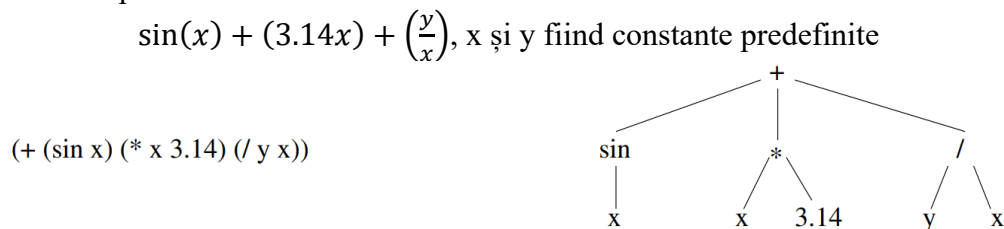


Fig. 3: Exemplu “S-expression” (stângă), și arbore de sintaxa asociată (dreaptă)^[7]

În GP, este recunoscut faptul că lungimea soluției nu poate fi cunoscută din început, deci numărul de geni (codone) trebuie să fie și el însuși evoluat.

Fiecare individ în cadrul unei populații în GP parcurge o anumită “viață” care constă în procese asemănătoare cu cele din natura: naștere, mutație, încrucișare (crossover) și reproducere. După inițializare primei populații fiecare individ participant parcurge ciclic următoarele procedee (fig. 4.).^{[4][4][10]}

1. **Selecție.** Reprezintă un proces de alegere celor mai potriviți indivizi (după funcția “fitness”) pentru a produce următoarea generație a populației. Sunt mai multe moduri de selecție, care se alege conform cerințelor problemei;
2. **Crossover.** Procedeu constă în combinație genelor părinților. Sunt mai multe variante posibile de a efectua încrucișare: într-un singur punct, în doua puncte etc;^[10]
3. **Mutație.** Acest proces este prevăzut pentru evitarea cazurilor când populația “cade” într-un minim local al funcției, neajungând la soluția optimă (minim global). Reprezintă o mică modificare aleatoare a valorilor câtorva gene ale părinților;^{[10][11]}
4. **Reproducție.** Reprezintă un procedeu, în urmă căruia creează o nouă generație pe bază tuturor pașilor făcuți înainte. După ce o nouă generația a fost creată ciclul se repetă.^[7] Condiții de ieșire din ciclu se vor discuta în subcapitolul 2.

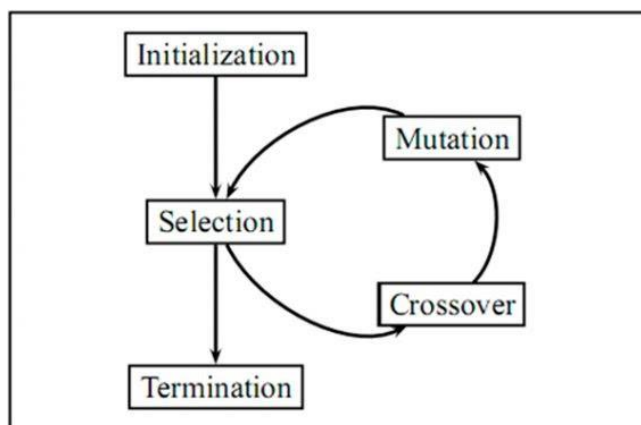


Fig. 4: Ciclu de viața unui individ în GP[9]

2. GRAMATICI EVOLUTIVE

GE reprezintă o formă de GP bazată pe formalismul unei gramatici. Pentru reprezentarea gramaticii se folosește un set de reguli numit Backus-Naur Forms (BNF). Acesta este un limbaj, propus de John Backus și Peter Naur ca o notăție formală pentru a descrie gramatica oricărui alt limbaj.^[13] Printre avantaje de aplicare Gramaticilor Evolutive putem menționa:^{[7][8][11]}

1. Aplicarea unui algoritm de “mappare” pentru transformarea unui șir “genotipic” binar sau de numere întregi produs de GP într-un program de ieșire “fenotipic”. Cu alte cuvinte, se realizează trecerea de la sintaxa de tip arbore (GP) la sintaxa de tip liniar;
2. GE are o arhitectură modulară (fig. 5.). Într-un algoritm, mașina de căutare, funcția obiectivă și gramatica sunt obiecte distincte, ce oferă mai mult spațiu pentru optimizări;
3. “Output” unui GE sunt date scrise în limbaje arbitrare programate în BNF.

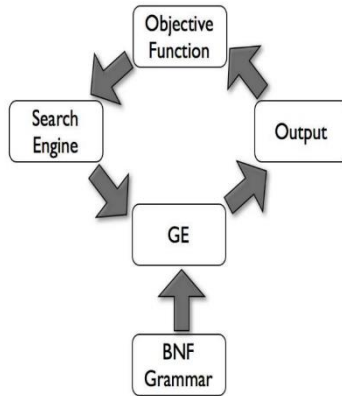


Fig. 5: Reprezentarea modularității designului de GE.^[8]

O gramatică BNF reprezintă un tuplu care conține 4 componente (fig. 5):^[8]

1. N - sunt simboluri non terminale. Ele sunt expresii care în urma procesului de mappare se vor transforma în T;
2. T - sunt simboluri terminale. “Output” unui program întotdeauna va conține numai aceste tipuri de simboluri;
3. P - sunt reguli de producție care descriu în totalitate ce se va întâmpla cu N în urma procesului de mappare. Un exemplu remarcabil este prezentat în fig. 7;
4. S - este expresie de început. Neapărat membru de N.

```

1  G = {N, T, P, S}
2  N = {<expr>, <biop>, <uop>, <bool>}
3  T = {and, or, xor, nand, not, true, false, (, )}
4  S = {<expr>}

```

Fig. 6: Reprezentare gramaticii din exemplu “Boolean Expression” (P este omis)^[7]

```

1  # "::=" - devine, "|" - logic "or"
2  # (n) - Numărul alegerii posibile
3  <syntax>      ::= { <rule> }
4  <rule>        ::= <identifier> "::=" <expression>
5  <expression>  ::= <term> { "|" <term> }
6  <term>        ::= <factor> { <factor> }
7  <factor>      ::= <identifier> (0)
8                | <quoted_symbol> (1)
9                | "(" <expression> ")" (2)
10               | "[" <expression> "]" (3)
11               | "{" <expression> "}" (4)
12 <identifier>  ::= <letter> { <letter> | <digit> }
13 <quoted_symbol> ::= ""{ any_character }""

```

Fig. 7: Gramatica BNF care descrie reguli de BNF^[13]

Proces de mappare se începe în moment când GP returnează o nouă generație. Începând cu primul individ al populației o regulă de calcul restului se aplică la fiecare codonul lui.

$$RULE = c \bmod r,$$

unde “c” - valoare de codon, iar “r” - numărul alegerilor posibile unui simbol non terminal.

De exemplu, în cazul prezentat în fig. 7, presupunem ca următorul termen N este $\langle \text{factor} \rangle$, $c = 10$, atunci făcând calculul $10 \bmod 5 = 0$, obținem că $\langle \text{factor} \rangle$ devine $\langle \text{identifîier} \rangle$. Ciclul se repetă, trecând la următoarea valoare de codon. Se repetă până la moment, când una sau mai multe dintre următoarele condiții sunt îndeplinite:^{[7][8]}

1. Programul este generat complet. Toți membrii de N s-au trecut prin proces de mappare și s-au transformat în T ;
2. Procesul de mappare a ajuns la sfârșitul genomului unui individ. Atunci se aplică operatorul de împachetare (wrapping) care pornește proces de re folosire genomului;
3. Se ajunge la limita numărului de iterații a operatorului de împachetare. În acest caz individ este asociat cu cea mai mică valoare a funcției “fitness”, ce duce la eliminarea sa din populație. A doua variantă este ca individul să se asocieze cu valoare prestabilită.

La sfârșit, când GP atinge limita generațiilor, el returnează un răspuns scris în limbaj arbitrar, conform gramaticii predefinite.

REPARTIȚIA WEIBULL. MODELAREA VITEZEI VÂNTULUI

1. FORMALISM WEIBULL

Repartiția Weibull este foarte des utilizată pentru modelarea datelor legate de viteza vântului. [1–22] În lucrarea de față sunt folosite Repartiția Weibull cu 2 parametrii și Metoda Maximei Verosimilități (MLM) pentru o aproximare cât mai exactă a parametrului de formă și a parametrului de scală.

Densitatea de probabilitate (PDF) a Repartiției Weibull este dată de relația (1):

$$X(\beta, c) = f(x) = \frac{\beta}{c^\beta} x^{\beta-1} \exp\left(-\left(\frac{x}{c}\right)^\beta\right), \quad \text{pentru } x > 0 \quad (1)$$

unde x este viteza vântului și $X(\beta, c)$ este o variabilă aleatoare, având parametrii β , respectiv, c ai Repartiției Weibull.

Distribuția cumulativă (CDF) a Repartiției Weibull este dată de relația (2):

$$F(x) = 1 - \exp\left(-\left(\frac{x}{c}\right)^\beta\right) \quad (2)$$

unde β este parametrul de formă (adimensional), c este parametrul de scală care se măsoară în unități de viteză a vântului ($m \cdot s^{-1}$).

În general, parametrul de formă β are valori cuprinse între 1.5 și 3, fiecare număr fiind atribuit unui anumit climat. Parametrul de scală c ia valori în funcție de β și este direct proporțional cu viteza medie a vântului.

În continuare, se utilizează Metoda Maximei Verosimilități pentru a estima parametrul de formă, respectiv, parametrul de scală. Valorile estimate se obțin rezolvând ecuația neliniară :

$$\frac{1}{\beta} - \left(\sum_{i=1}^n x_i^\beta \ln x_i\right) \left(\sum_{i=1}^n x_i^\beta\right)^{-1} + \frac{1}{n} \sum_{i=1}^n \ln x_i = 0 \quad (3)$$

$$c = \left(\frac{1}{n} \sum_{i=1}^n x_i^\beta\right)^{\frac{1}{\beta}} \quad (4)$$

Soluțiile ecuației (3) se determină cu Metoda Newton, aplicată iterativ, iar rezultatele vor fi valorile approximate ale parametrilor Repartiției Weibull, respectiv $\hat{\beta}$ și \hat{c} .

STUDIU DE CAZ

În prezentul studiu de caz se consideră două zone geografice din România, în care se vor analiza viteza vântului și comportamentul parametrilor Repartiției Weibull pe durata unei zile. Scopul acestui studiu este de a estima producția de energie, respectiv rolul centralelor eoliene în estimarea cererii de energie, pentru două tipuri diferite de așezări umane, respectiv două climate diferite.

Se iau în considerare Comuna Gorbănești din județul Botoșani, și orașul Timișoara. Comuna Gorbănești este situată în extremitatea nord-estică a României, în centrul județului Botoșani, zona fiind caracterizată de un relief deluros.^[14] Clima este temperat-continentală, fiind influențată puternic de masele de aer din estul continentului^[15] (influențe extrem – continentale). Valorile pe care viteza vântului le-a înregistrat în comuna Gorbănești, măsurate în data 5.05.2020, sunt reprezentate în Tabelul 1, , conform site-ului www.windfinder.com.

Orașul Timișoara este localizat în sud – estul Câmpiei Panonice, în partea de vest a României. Climatul este temperat continental moderat, având influențe submediteraneene^[16]. Valorile înregistrate ale vitezei vântului la data 5.05.2020, în Timișoara sunt redată în Tabelul 2, conform site-ului www.windfinder.com.

Tabelul 1. Viteza vântului în comuna Gorbănești

Viteza Vântului	Timp de înregistrare
5.14	00:00
3.09	06:00
3.09	12:00
2.06	18:00

Tabelul 2. Viteza vântului în orașul Timișoara

Viteza Vântului	Timp de înregistrare
0	00:00
1.03	06:00
2.06	12:00
2.06	18:00

În fig. 8 este reprezentată variația în timp a vitezei vântului pentru Botoșani, respectiv, Timișoara.

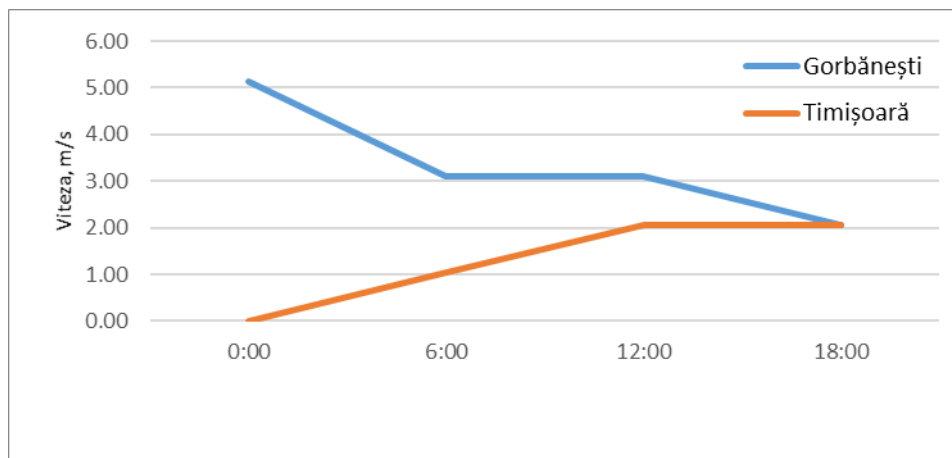


Fig. 8: Variația vitezei vântului în funcție de timp

Valorile din Tabelele 1 și 2 sunt introduse în formula (4), pentru fiecare valoare a parametrului de formă $\beta \in [1.5, 3]$, obținând parametrul de scală c . Calculele sunt realizate cu ajutorul unui program în Python ilustrat în fig. 9.

```

1 import numpy as np
2
3 def scale_param(vv, timp, beta):
4     for i in range(len(vv)):
5         print(f"Wind speed: {vv[i]}, Inregistrat: {timp[i]}")
6     for i in range(len(beta)):
7         summ = 0
8         shape_param = beta[i]
9         print(f"\nShape parameter (beta) = {round(shape_param, 2)}")
10        for j in range(len(vv)):
11            xjpowk = vv[j] ** shape_param
12            summ += xjpowk
13        c = (1/len(vv) * summ) ** (1/shape_param)
14        shape_param = str(shape_param)[:4]
15        c = str(c)[:6]
16        print(f"Scale parameter (c) = {c}")
17
18 vv = [5.14, 3.09, 3.09, 2.06]
19 timp = ["00:00", "06:00", "12:00", "18:00"]
20 print("Date pentru ziua de 5 mai, 2020:")
21 beta = np.arange(1.5, 3.01, 0.01)
22 scale_param(vv, timp, beta)
23

```

Output:

```

D:\S3N1CH\Soft\Python\python.exe "E:/S
Date pentru ziua de 5 mai, 2020:
Wind speed: 5.14, Inregistrat: 00:00
Wind speed: 3.09, Inregistrat: 06:00
Wind speed: 3.09, Inregistrat: 12:00
Wind speed: 2.06, Inregistrat: 18:00

Shape parameter (beta) = 1.5
Scale parameter (c) = 3.4357

Shape parameter (beta) = 1.51
Scale parameter (c) = 3.4375

Shape parameter (beta) = 1.52
Scale parameter (c) = 3.4393

Shape parameter (beta) = 1.53
Scale parameter (c) = 3.4411

Shape parameter (beta) = 1.54
Scale parameter (c) = 3.4430

Shape parameter (beta) = 1.55
Scale parameter (c) = 3.4448

```

Fig. 9: Secvența de cod pentru determinarea parametrului de scală

În secvența de cod, vectorul $\beta \in [1.5, 3]$ este parcurs cu pasul 0.01 pentru acuratețe, deoarece se ia în considerare identificarea celei mai potrivite valori pentru climatul propus. Valorile parcurse pentru parametrul de formă β , și calculate pentru parametrul de scală c sunt expuse în Tabelul 3 pentru Comuna Gorbănești, respectiv Tabelul 4 pentru orașul Timișoara.

Tabelul 3. Valorile lui β și c , pentru Gorbănești

	Parametru de formă β	Parametru de scală c	
β_1	1.50	3.4357	c_1
β_2	1.51	3.4375	c_2
...
β_{152}	3.00	3.7056	c_{152}

Tabelul 4. Valorile lui β și c , pentru Timișoara

	Parametru de formă β	Parametru de scală c	
β_1	1.50	1.1598	c_1
β_2	1.51	1.1592	c_2
...
β_{152}	3.00	1.5480	c_{152}

Atunci când valorile lui β sunt mari, vântul este mai stabil decât pentru valorile mici, acestea din urmă

Valorile obținute se pot prelucra cu ajutorul Gramaticilor Evolutive (GE) pentru determinarea parametrului de scală c , în funcție de timp și variabile introduse de mediu. Pentru exemplificarea procesului am considerat schema logică reprezentată în fig. 10^[16], și am introdus variabilele I, radiația solară, T, temperatura ambientală, și P, presiunea atmosferică. Estimarea

parametrului de scală este realizată în funcție de timp, t , și de variabilele $I(t-1)$, $T(t-1)$, și $P(t-1)$.

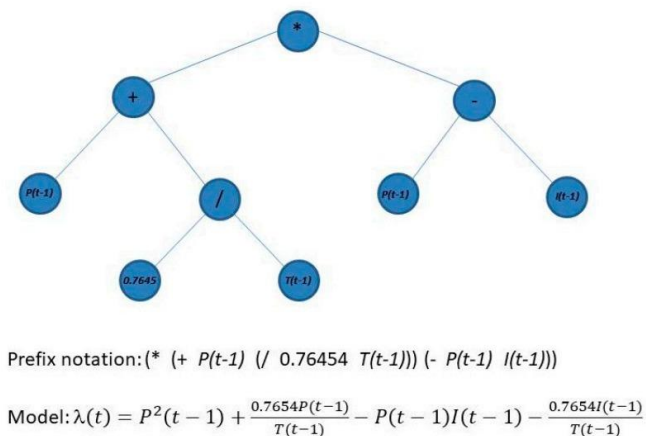


Fig. 10: Schema logică a algoritmului GP

Secvența de cod bazată pe formalismul Gramaticilor Evolutive care va estima expresia parametrului de scală c este redată în fig 11.

```

1  <c> ::= "c = c(t) = " <c_polinom>
2  <operator> ::= "+" | "-" | "*" | "/" | "^"
3  <caracter> ::= "I" | "T" | "P"
4  <integ> ::= [0-9]
5             | <integ> <integ>
6  <real> ::= "0." <integ>
7  <constanta> ::= <integ> | <real>
8  <functie> ::= <constanta> "*" <functie>
9             | <caracter> "(t-1)"
10            | <caracter> "^" <constanta> "(t-1)"
11 <monom> ::= "(" <monom> <operator> <monom> ")"
12            | <functie>
13 <c_polinom> ::= <monom> <operator> <monom>
14

```

Fig. 11: Secvența de cod a gramaticii (BNF)

Exemple de „output” generate de algoritmul din fig 11.

$$1) c = c(t) = 3 * I(t-1) * ((I^{0.6}(t-1) - T(t-1)) - I^5(t-1))$$

$$2) c = c(t) = (I(t-1)^{T(t-1)})^P(t-1)$$

Pornind de la secvența de cod din fig. 11, se poate realiza un nou program folosind Programarea Genetică (GP) pentru a selecta automat cel mai potrivit output pentru forma parametrului de scală.

Pentru a modela producția de energie eoliană, conform Asociației pentru Vânt, puterea unei turbine eoliene poate fi aproximată prin relația (5):

$$P = 0.5 A \rho C_p N_b N_g (ws)^3 \quad (5)$$

unde P este puterea motorului (Watt), A este suprafața rotorului expusă la vânt (m/s), ρ este densitatea aerului, ws este viteza vântului, C_p este coeficientul de performanță, N_g este eficiența generatorului, N_b este eficiența rulmentului.

Pe baza caracteristicilor rotorului turbinei se poate realiza simularea producției de energie, utilă pentru estimarea cererii de energie.

CONCLUZII

Concluzionând, centralele eoliene au un rol important în estimarea cererii de energie. Pe baza implementării unui algoritm GE la secvența de cod expusă în studiul de caz se poate estima valoare exactă a parametrului de formă, respectiv de scală.

Fiecare climat se poate asocia unei valori specifice a parametrului de formă, ceea ce înseamnă că Gramatica Evolutivă va progresa într-un mod unic pentru fiecare climat și așezare umană.

Cu ajutorul Repartiției lui Weibull se poate modela distribuția vitezei vântului, urmând să fie simulată valoarea producției de energie.

REFERINȚE

- [1] - Figura 1, https://www.researchgate.net/figure/Structure-diagram-of-micro-grid_fig1_270064845
- [2] – Micro Grid - un răspuns la noile provocări ale tehnologiei, <https://controlengrussia.com/otraslevye-resheniya/microgrid/>
- [3] - What is the difference between a microgrid and a smartgrid, https://www.researchgate.net/post/What_is_the_difference_between_a_microgrid_and_a_smartgrid
- [4] - Charles Darwin, https://en.wikipedia.org/wiki/Charles_Darwin
- [5] - Notății BNF și CF-gramatici, <http://csaa.ru/bnf-notaciya-i-ks-grammatiki/>
- [6] - Backus-Naur Form (BNF) notation, <http://matt.might.net/articles/grammars-bnf-ebnf/>
- [7] - Anthony Brabazon, Michael O'Neill, Biologically Inspired Algorithms for Financial Modelling, Germany, Springer-Verlag Berlin Heidelberg, 2006
- [8] - Ian Dempsey, Michael O'Neill and Anthony Brabazon, Foundations in Grammatical Evolution for Dynamic Environments, Springer-Verlag Berlin Heidelberg, 2009
- [9] - The language of languages, <http://matt.might.net/articles/grammars-bnf-ebnf/>
- [10] - A gentle introduction to genetic algorithm, <https://medium.com/the-z/a-gentle-introduction-to-genetic-algorithm-30695a6084c2>
- [11] - Introduction to optimization with Genetic Algorithm, <https://towardsdatascience.com/introduction-to-optimization-with-genetic-algorithm-2f5001d9964b>
- [12] - Survival of the Fittest Variation: Evolutionary Algorithms in Optimization, <https://cxl.com/blog/evolutionary-algorithms-optimization/>
- [13] - What is BNF notation, <http://cui.unige.ch/isi/bnf/AboutBNF.html>
- [14] - Județul Botoșani – prezentare, <http://www.isubotosani.ro/botosani.html>
- [15] – Botoșani, <https://ro.wikipedia.org/wiki/Boto%C8%99ani#Clima>
- [16] - Long-term Estimation of Wind Power by Probabilistic Forecast Using Genetic Programming, Mónica Borunda 1, Katya Rodríguez-Vázquez 2 , Raul Garduno-Ramirez 3 , Javier de la Cruz-Soto 1 , Javier Antunez-Estrada 3 and Oscar A. Jaramillo 4, Published: 13 April 2020