

# KLASTEROVANJE TEKSTUALNIH DOKUMENATA POBOLJŠANOM HIBRIDNOM METODOM METAHEURISTIKE I ALGORITMOM NENADGLEDANOG MAŠINSKOG UČENJA

*By* Timea Bezdan



<sup>12</sup>UNIVERZITET SINGIDUNUM

BEOGRAD

DEPARTMAN ZA POSLEDIPLOMSKE STUDIJE

DOKTORSKA DISERTACIJA

---

KLASTEROVANJE TEKSTUALNIH DOKUMENATA  
POBOLJŠANOM HIBRIDNOM METODOM  
METAHEURISTIKE I ALGORITMOM  
NENADGLEDANOG MAŠINSKOG UČENJA

---

*Mentor:* prof. dr Nebojša

BAČANIN-DŽAKULA

*Student:* Timea BEZDAN

*Broj indeksa:* 2021460016

Beograd, 2024.



GRADUATE SCHOOL

PHD THESIS

---

**TEXT DOCUMENT CLUSTERING BY IMPROVED  
HYBRID METAHEURISTIC METHOD AND  
UNSUPERVISED MACHINE LEARNING ALGORITHM**

---

*Mentor:*

prof. <sup>5</sup>Nebojša

BAČANIN-DŽAKULA, PhD

*Candidate:*

Timea BEZDAN

Belgrade, 2023.

**Mentor:**

Prof. dr Nebojša Bačanin-Džakula, redovni profesor  
Univerzitet Singidunum, Beograd

**Članovi komisije:**

Prof. dr Miodrag Živković, redovni profesor  
Univerzitet Singidunum, Beograd

*Predsednik komisije*

Prof. dr Boško Nikolić, redovni profesor  
Univerzitet u Beogradu, Elektrotehnički fakultet, Beograd

*Član komisije*

## Zahvalnica

*Zahvaljujem se Bogu svemogućemu za pokazivanje pravaca, snagu, inspiraciju, strpljenje i istrajnost na putu realizacije ove doktorske disertacije sa zajedništvom Duha Svetoga.*

19

*“Omni enim habenti dabitur et abundabit ei autem qui non habet et quod videtur habere auferetur ab eo.”*

– Matthæus 25:29 - Parabole des talents

20

*“Mert mindenkinek, akinek van, adatik, és bővelkedni fog; attól pedig, akinek nincs, még az is elvételik, amije van.”*

– Máté 25:29 - A talentumokról szóló példabeszéd

# Klasterovanje tekstualnih dokumenata poboljšanom hibridnom metodom metaheuristike i algoritmom nenadgledanog mašinskog učenja

**Sažetak** – U poslednje vreme, usled rapidnog rasta veb stranica, društvenih medija i drugih savremenih digitalnih aplikacija, količina tekstualnih podataka postaje enormna. Zbog velikog obima heterogenih informacija, nestruktuiranog formata tekstualnih podataka, izdvajanje relevantnih informacija i njihova analiza postaju veoma izazovni i predstavljaaju značajnu oblast naučnog istraživanja. Klasterovanje tekstualnih dokumenata je krucijalni proces za pronalaženje i ekstrakciju informacija kao i za organizaciju dokumenata, stoga klasterovanje je značajan zadatak u radu sa ogromnim količinama tekstualnih podataka. Klasterovanje tekstualnih dokumenata je proces rudarenja teksta koji deli skup dokumenata zasnovanih na tekstu u međusobno isključive klasterove na način da su dokumenti unutar iste grupe imaju veliki stepen sličnosti, dok se dokumenti iz različitih klastera razlikuju po sadržaju. Jedan od najvećih izazova u klasterovanju teksta je particionisanje kolekcije tekstualnih podataka merenjem relevantnosti sadržaja u dokumentima. Jedan od često korišćenih algoritama za klasterovanje je algoritam K-srednjih vrednosti koji je jednostavan i lak za implementaciju, međutim ima određenih nedostataka, kao što je problem inicijalizacije i zaglavljivanje u lokalnom optimumu što iziskuje potrebu razvoja nove inovativne tehnike koja efikasno rešava problem klasterovanja tekstualnih dokumenata. Baveći se ovim problemom, glavni doprinos ove disertacije je novi poboljšani hibridni algoritam koji je baziran na inteligenciji roja i kombinovana sa algoritmom K-srednjih vrednosti za klasterovanje tekstualnih dokumenata. Eksplicitno, algoritam voćne mušice je hibridizovan sa algoritmom svica, dodatno u proces pretraživanja algoritma svica je implementirana strategija dinamičnog prilagođavanja adaptivnog parametra  $\alpha$ . Radi bržeg i efikasnijeg pronalaženja optimalnijeg rešenja, u algoritam je ugrađen i popularan delotvorni koncept učenje zasnovanog na opoziciji. U cilju dokazivanja efikasnosti algoritma, kao standardna procedura kod bilo kog novouvedenog metaheurističkog algoritma, benčmark test funkcije se koriste za kritičko razmatranje procene performanse i merenje kvaliteta. Algoritmi koji su uspešni u rešavanju numeričkih optimizacionih problema, smatraju se uspešnim i efikasnim

metodama za rešavanje problema iz realnog sveta. Skladno tome, predloženi poboljšani hibridni metod voćne mušice je prvo validiran na deset modernih neograničenih CEC 2019 benčmark funkcija. Zatim se predloženi metod kombinovan sa algoritmom K-srednjih vrednosti i primenjen za problem klasterovanja tekstualnih dokumenata, korišćenjem šest standardnih benčmark tekstualnih skupova podataka. Dobijeni rezultati simulacije na neograničenim funkcijama potvrđuju pouzdanost i potencijal fuzije kvalitetnih komponenata u novom predloženom algoritmu. Statistički pokazatelji simulacije i komparativna analiza sa drugih deset metaheurističkih pristupa potkrepljuje pogodnost predloženog hibridnog pristupa koji ima snažnu sposobnost u pronalaženju globalnog optimalnog rešenja u razumnom vremenu. U drugoj simulaciji, algoritam je primenjen na problem klasterovanja tekstualnih dokumenata, dobijeni rezultati eksperimenta klasterovanja i komparativna analiza pokazuju da je predloženi pristup robustan i superioran u odnosu na druge najsavremenije metaheurističke i nemetaheurističke metode.

**Ključne reči:** klasterovanje tekstualnih dokumenata, mašinsko učenje, K-srednjih vrednosti, metaheuristički algoritmi, algoritam optimizacije voćne mušice.

# Text document clustering by improved hybrid metaheuristic method and unsupervised machine learning algorithm

<sup>29</sup>**Abstract** – Recently, due to the rapid growth of web pages, social media, and other modern digital applications, the amount of textual data has become enormous. Due to the large volume of heterogeneous information, the unstructured format of textual data, the extraction of relevant information, and its analysis become very challenging and represent an important area of scientific research. Text document clustering is a crucial process for finding and extracting information as well as for organizing documents, therefore clustering is a significant task in working with huge amounts of textual data. Text document clustering represents a text mining process that divides a set of text-based documents into mutually exclusive clusters in such a way that documents within the same group have a high degree of similarity, while documents from different clusters vary in content. One of the biggest challenges in text clustering is partitioning a collection of text data by measuring the relevance of content in the documents. One of the frequently used clustering algorithms is the K-means algorithm, which is simple and easy to implement, but it has certain drawbacks, such as the initialization problem and getting stuck in the local optimum, which necessitates the development of a new innovative technique that effectively solves the problem of text document clustering. Addressing this problem, the main contribution of this dissertation is a new improved hybrid algorithm based on swarm intelligence combined with the K-means algorithm for text document clustering. Explicitly, the fruit-fly optimization algorithm is hybridized with the firefly algorithm, and additionally, in the search process of the firefly algorithm, a strategy of dynamic adaptive parameter  $\alpha$  is implemented. In order to find an optimal solution faster and more efficiently, the popular effective concept of opposition-based learning is incorporated into the algorithm. To prove the efficiency of the algorithm, as a standard procedure of any newly introduced metaheuristic algorithm, benchmark test functions are used to critically consider the performance evaluation and quality measurement. Algorithms that are successful in solving numerical optimization problems are considered successful and efficient methods for solving real-world problems.

Accordingly, the proposed hybrid fruit-fly algorithm method is first validated on ten modern unconstrained CEC 2019 benchmark functions. Then, the proposed method was combined with the K-means algorithm and applied to the problem of clustering text documents by using six standard benchmark text datasets. The obtained simulation results on unconstrained functions confirm the reliability and potential of the fusion of quality components in the new proposed algorithm. The statistical indicators of the simulation and the comparative analysis with the other ten metaheuristic approaches support the reliability of the proposed hybrid approach which has a strong ability to find the global optimal solution in a reasonable time. In the second simulation, the algorithm was applied to the text document clustering problem. The results of the clustering experiment and the comparative analysis show that the proposed approach is robust and superior in comparison to other state-of-the-art metaheuristic and non-metaheuristic methods.

**Keywords:** text document clustering, machine learning, K-means, metaheuristic algorithms, fruit-fly optimization algorithm.

## Spisak slika

|    |                                                                                                                                         |    |
|----|-----------------------------------------------------------------------------------------------------------------------------------------|----|
| 1  | Dijagram toka opšteg dizajna evolutivnih algoritama . . . . .                                                                           | 30 |
| 2  | Dijagram toka opšteg dizajna algoritama inteligencije roja algoritama                                                                   | 32 |
| 3  | Veštačka inteligencija, mašinsko učenje i duboko učenje . . . . .                                                                       | 34 |
| 4  | Ilustracija procesa itarativnog traženja hrane u algoritmu voćne mušice                                                                 | 59 |
| 5  | Dijagram toka originalnog algoritma voćne mušice . . . . .                                                                              | 61 |
| 6  | Dijagram toka algoritma svica . . . . .                                                                                                 | 64 |
| 7  | Primer OBL-a u jednodimenzijonalnom prostoru . . . . .                                                                                  | 66 |
| 8  | Primer OBL-a u dvodimenzijonalnom prostoru ( $i = 0$ ) . . . . .                                                                        | 66 |
| 9  | Primer OBL-a u dvodimenzijonalnom prostoru ( $i = 1$ ) . . . . .                                                                        | 66 |
| 10 | Primer OBL-a u dvodimenzijonalnom prostoru ( $i = 2$ ) . . . . .                                                                        | 67 |
| 11 | Primer polaznog skupa podataka . . . . .                                                                                                | 67 |
| 12 | Primer sa incijalnim težištima (centroidima) klastera . . . . .                                                                         | 68 |
| 13 | Primer prvog koraka u iteraciji - dodeljivanje uzoraka klasterima . . .                                                                 | 69 |
| 14 | Primer drugog koraka u iteraciji - pomeranje centroida klastera . . .                                                                   | 69 |
| 15 | Primer konvergencije algoritma koji zaustavlja proces . . . . .                                                                         | 70 |
| 16 | Dijagram toka algoritma K-srednjih vrednosti . . . . .                                                                                  | 70 |
| 17 | Dijagram toka predloženog HEFFF algoritma . . . . .                                                                                     | 73 |
| 18 | Dijagram toka procesa obrade ulaznih tekstualnih podataka . . . . .                                                                     | 75 |
| 19 | Dijagram toka predloženog HEFFF algoritma u kombinaciji sa algo-<br>ritmom K-srednjih vrednosti za klasterovanje tekstualnih dokumenata | 77 |
| 20 | Dijagram toka klasterovanja tekstualnih dokumenata . . . . .                                                                            | 79 |
| 21 | 3D grafikon dvodimenzionalne Rastrigin funkcije . . . . .                                                                               | 84 |
| 22 | Kontur grafikon dvodimenzionalne Rastrigin funkcije . . . . .                                                                           | 85 |
| 23 | 3D grafikon dvodimenzionalne Griewank funkcije . . . . .                                                                                | 85 |
| 24 | Kontur grafikon dvodimenzionalne Griewank funkcije . . . . .                                                                            | 86 |
| 25 | 3D grafikon dvodimenzionalne Weierstrass funkcije . . . . .                                                                             | 87 |
| 26 | Kontur grafikon dvodimenzionalne Weierstrass funkcije . . . . .                                                                         | 87 |
| 27 | 3D grafikon dvodimenzionalne Schwefel funkcije . . . . .                                                                                | 88 |
| 28 | Kontur grafikon dvodimenzionalne Schwefel funkcije . . . . .                                                                            | 88 |
| 29 | 3D grafikon dvodimenzionalne Schaffer funkcije . . . . .                                                                                | 89 |
| 30 | Kontur grafikon dvodimenzionalne Schaffer funkcije . . . . .                                                                            | 90 |

|    |                                                                                                   |     |
|----|---------------------------------------------------------------------------------------------------|-----|
| 31 | 3D grafikon dvodimenzionalne Happy Cat funkcije . . . . .                                         | 91  |
| 32 | Kontur grafikon dvodimenzionalne Happy Cat funkcije . . . . .                                     | 91  |
| 33 | 3D grafikon dvodimenzionalne Ackley funkcije . . . . .                                            | 92  |
| 34 | Kontur grafikon dvodimenzionalne Ackley funkcije . . . . .                                        | 92  |
| 35 | Grafikoni konvergenције HEFFF i FFO metode na CEC 2019 test<br>funkcijama . . . . .               | 98  |
| 36 | Uporedni prikaz rezultata stope greške HEFFF metode i ostalih kom-<br>parabilnih metoda . . . . . | 106 |
| 37 | Uporedni prikaz rezultata tačnosti HEFFF metode i ostalih kompara-<br>bilnih metoda . . . . .     | 106 |
| 38 | Uporedni prikaz rezultata preciznosti HEFFF metode i ostalih kom-<br>parabilnih metoda . . . . .  | 106 |
| 39 | Uporedni prikaz rezultata odziva HEFFF metode i ostalih kompara-<br>bilnih metoda . . . . .       | 107 |
| 40 | Uporedni prikaz rezultata F-mere HEFFF metode i ostalih kompara-<br>bilnih metoda . . . . .       | 107 |
| 41 | Uporedni prikaz rezultata čistoće HEFFF metode i ostalih kompara-<br>bilnih metoda . . . . .      | 107 |
| 42 | Uporedni prikaz rezultata entropije HEFFF metode i ostalih kompa-<br>rabilnih metoda . . . . .    | 108 |

## Spisak tabela

|    |                                                                                                                  |     |
|----|------------------------------------------------------------------------------------------------------------------|-----|
| 1  | Kategorije tradicionalnih algoritama za klasterovanje i primeri svakog tipa . . . . .                            | 37  |
| 2  | Kontrolni parametri . . . . .                                                                                    | 93  |
| 3  | Statistički rezultati simulacije na neograničenim CEC 2019 funkcijama                                            | 94  |
| 4  | Fridmanovo rangiranje uporedivih metoda na CEC 2019 test funkcijama                                              | 96  |
| 5  | Fridmanov i Iman-Davenportov statistički test rezultati ( $\alpha = 0.05$ ) . .                                  | 97  |
| 6  | Rezultati Holmove procedure . . . . .                                                                            | 97  |
| 7  | Karakteristika skupa tekstualnih podataka korišćene za simulacije klasterovanja tekstualnih dokumenata . . . . . | 100 |
| 8  | Inicijalni vrednosti kontrolnih parametara . . . . .                                                             | 102 |
| 9  | Rezultati klasterovanja tekstualnih podataka na CSTR skupu podataka                                              | 103 |
| 10 | Rezultati klasterovanja tekstualnih podataka na <i>20 Newsgroups</i> skupu podataka . . . . .                    | 103 |
| 11 | Rezultati klasterovanja tekstualnih podataka na Tr12 skupu podataka                                              | 104 |
| 12 | Rezultati klasterovanja tekstualnih podataka na Tr41 skupu podataka                                              | 104 |
| 13 | Rezultati klasterovanja tekstualnih podataka na Wap skupu podataka                                               | 104 |
| 14 | Rezultati klasterovanja tekstualnih podataka na Classic 4 skupu podataka . . . . .                               | 105 |

## Algoritmi

|   |                                                                                                                               |    |
|---|-------------------------------------------------------------------------------------------------------------------------------|----|
| 1 | Opšti pseudokod evolutivnih algoritama . . . . .                                                                              | 30 |
| 2 | Opšti pseudokod algoritma inteligencije roja . . . . .                                                                        | 31 |
| 3 | Pseudokod originalnog FFO algoritma . . . . .                                                                                 | 61 |
| 4 | Pseudokod originalnog FA algoritma . . . . .                                                                                  | 64 |
| 5 | Pseudokod algoritma K-srednjih vrednosti . . . . .                                                                            | 71 |
| 6 | Pseudokod predloženog HEFFF algoritma . . . . .                                                                               | 74 |
| 7 | Pseudokod predloženog HEFFF algoritma sa algoritmom K-srednjih<br>vrednosti za klasterovanje tekstualnih dokumenata . . . . . | 78 |

## Skraćenice

- <sup>35</sup>**ABC** Artificial bee colony algorithm
- ACO** Ant colony optimization
- AI** Artificial intelligence
- ANN** Artificial neural network
- ApFA** Firefly algorithm with adaptive control parameters - ApFA
- AUC** Area under the curve
- BBO** Biogeography based optimization
- BGWO** Binary <sup>54</sup>grey wolf optimizer
- CEC** Congress on evolutionary computation
- CMAES** Covariance matrix adaptation evolution strategy
- <sup>37</sup>**COA** Coyote optimization algorithm
- CS** Cuckoo search
- CSO** Cat swarm optimization
- CSTR** Centre for Speech Technology Research
- DBSCAN** Density-based clustering <sup>24</sup>algorithm
- DE** Differential evolution
- EA** Evolutionary algorithm
- EHO** Elephant herding optimization
- EHOI** Improved elephant herding optimization
- EMM** Expected maximization method
- FA** Firefly algorithm
- FCM** Fuzzy C-means

**FCPSO** Hybrid of FCM and PSO algorithm

**FFO** Fruit-fly optimization

**FOAGRNN** <sup>17</sup> Fruit fly algorithm optimized general regression neural network

**GA** Genetic algorithm

**GF-CLUST** Gravity firefly clustering

**GMP** Geometric programming problem

<sup>50</sup>  
**GOA** Grasshopper optimization algorithm

**GRNN** General regression neural network

**GSA** Gravitational search <sup>7</sup> algorithm

**GWO** Grey wolf optimizer

**HC** Hill climbing algorithm

**HEFFF** Hybrid enhanced fruit-fly firefly

**HS** Harmony search

**IP** Integer programming

**JO** JAYA optimization

**KHA** Krill herd algorithm

**KHM** K-harmonic means

**KPSO** Hybrid of K-Means and PSO algorithm

**LBMVO** Link-based multiverse optimizer

**LP** Linear programming

**LSI** Latent semantic indexing

**MA** Memetic algorithm

**MFO** Moth flame optimization

32

**MILP** Mixed-integer linear programming

**MINLP** Mixed-integer nonlinear programming

**ML** Machine learning

**MR** Multiple regression

**MVO** Multiverse optimizer

**NIPS 2015** Neural Information Processing Systems 2015

**NLP** Nonlinear programming problem

**NP** Nondeterministic polynomial time

**OBL** Opposition-based learning

**PCA** Principal component analysis

**pGSCM** Practical general stochastic clustering method

**PGWO** Particle grey wolf optimizer

44

**PSO** Particle swarm optimization

**QP** Quadratic programming

**RL** Reinforcement learning

56

**SCA** Sine cosine algorithm

**SI** Swarm intelligence

**SOA** seagull optimization algorithm

16

**SSA** Salp swarm algorithm

**SSO** Social spider optimization

**SVD** Singular value decomposition

57

**TDC** Text document clustering

**TF-IDF** Term Frequency-Inverse Document Frequency

**TSP** Traveling salesman problem

**VSM** Vector space model

**WOA** Whale optimization algorithm

# Sadržaj

|                                                                |           |
|----------------------------------------------------------------|-----------|
| Spisak slika                                                   | 7         |
| Spisak tabela                                                  | 9         |
| Algoritmi                                                      | 10        |
| Skraćenice                                                     | 11        |
| <b>5</b><br><b>1 Uvod</b>                                      | <b>18</b> |
| 1.1 Predmet i problem istraživanja                             | 19        |
| 1.2 Cilj istraživanja                                          | 20        |
| 1.3 Hipoteze u istraživanju                                    | 21        |
| 1.4 Metode istraživanja                                        | 21        |
| 1.5 Naučni doprinos istraživanja                               | 22        |
| 1.6 Struktura rada                                             | 22        |
| <b>2 Optimizacija i kategorizacija optimizacionih problema</b> | <b>24</b> |
| 2.1 Koncept optimizacije i matematička formulacija             | 24        |
| 2.2 Računska složenost                                         | 27        |
| 2.3 Algoritmi stohastičke aproksimacije                        | 28        |
| 2.4 Metaheuristički algoritmi                                  | 28        |
| 2.5 Evolutivni algoritmi                                       | 29        |
| 2.6 Algoritmi inteligencije roja                               | 31        |
| 2.7 Hibridni metaheuristički algoritmi                         | 31        |
| <b>3 Mašinsko učenje</b>                                       | <b>34</b> |
| 3.1 Nadgledano učenje                                          | 35        |
| 3.2 Nenadgledano učenje                                        | 36        |
| 3.2.1 Klasterovanje                                            | 37        |
| 3.2.2 Mere sličnosti koje se koriste kod klasterovanja         | 38        |
| 3.3 Polunadgledano učenje                                      | 40        |
| 3.4 Učenje sa pojačanjem                                       | 40        |

|                                                                                                                                                        |               |
|--------------------------------------------------------------------------------------------------------------------------------------------------------|---------------|
| <b>4 Pregled literature klasterovanja tekstualnih dokumenata</b>                                                                                       | <b>41</b>     |
| 4.1 Klasterovanje tekstualnih dokumenata algoritmom optimizacije roja<br>čestica . . . . .                                                             | 41            |
| 4.2 Klasterovanje tekstualnih dokumenata genetskim algoritmom . . . . .                                                                                | 43            |
| 4.3 Klasterovanje tekstualnih dokumenata algoritmom diferencijalne evo-<br>lucije . . . . .                                                            | 46            |
| 4.4 Klasterovanje tekstualnih dokumenata algoritmom optimizacije sivog<br>vuka . . . . .                                                               | 47            |
| 4.5 Klasterovanje tekstualnih dokumenata algoritmom potrage za kukavicom                                                                               | 47            |
| 4.6 Klasterovanje tekstualnih dokumenata algoritmom svica . . . . .                                                                                    | 48            |
| 4.7 Klasterovanje tekstualnih dokumenata algoritmom krda krila . . . . .                                                                               | 49            |
| 4.8 Klasterovanje tekstualnih dokumenata algoritmom optimizacije dru-<br>štvenih pauka . . . . .                                                       | 50            |
| 4.9 Klasterovanje tekstualnih dokumenata algoritmom optimizacije kitova                                                                                | 51            |
| 4.10 Klasterovanje tekstualnih dokumenata algoritmom optimizacije kolo-<br>nije mrava . . . . .                                                        | 51            |
| 4.11 Klasterovanje tekstualnih dokumenata algoritmom traženje harmonije                                                                                | 52            |
| 4.12 Klasterovanje tekstualnih dokumenata optimizatorom multiverzuma .                                                                                 | 53            |
| 4.13 Drugi metaheuristički algoritmi za klasterovanje tekstualnih dokumenata                                                                           | 54            |
| <br><b>5 Klasterovanje tekstualnih dokumenata poboljšanom hibridnom me-<br/>todom metaheuristike i algoritmom nenadgledanog mašinskog uče-<br/>nja</b> | <br><b>56</b> |
| 5.1 Problem klasterovanja tekstualnih dokumenata . . . . .                                                                                             | 56            |
| 5.2 Algoritam optimizacije voćne mušice . . . . .                                                                                                      | 58            |
| 5.2.1 Prednosti i nedostaci originalnog algoritma voćne mušice . . .                                                                                   | 62            |
| 5.3 Algoritam svica . . . . .                                                                                                                          | 62            |
| 5.4 Učenje zasnovano na opoziciji . . . . .                                                                                                            | 63            |
| 5.5 K-srednjih vrednosti . . . . .                                                                                                                     | 67            |
| 5.6 Hibridna poboljšana voćna mušica i svitac . . . . .                                                                                                | 70            |
| 5.7 Klasterovanje tekstualnih dokumenata poboljšanom hibridnom meto-<br>dom HEFFF i algoritmom K-srednjih vrednosti . . . . .                          | 73            |

|          |                                                                                                     |            |
|----------|-----------------------------------------------------------------------------------------------------|------------|
| <b>6</b> | <b>Eksperimentalni rezultati</b>                                                                    | <b>80</b>  |
| 6.1      | Eksperimenti na CEC 2019 benčmark funkcijama . . . . .                                              | 80         |
| 6.1.1    | Definicija CEC 2019 test funkcija . . . . .                                                         | 81         |
| 6.1.2    | Rezultati na neograničenim funkcijama . . . . .                                                     | 91         |
| 6.2      | Eksperimenti klasterovanja tekstualnih dokumenata . . . . .                                         | 97         |
| 6.2.1    | Tekstualni skupovi podataka korišćeni u simulacijama klasterovanja tekstualnih dokumenata . . . . . | 99         |
| 6.2.2    | Mere evaulacije kvaliteta klasterovanja . . . . .                                                   | 100        |
| 6.2.3    | Rezultati i analiza eksperimenta klasterovanja . . . . .                                            | 102        |
| <b>7</b> | <b>Zaključak</b>                                                                                    | <b>109</b> |
| <b>8</b> | <b>Literatura</b>                                                                                   | <b>112</b> |

# 1 Uvod

Klasterovanje tekstualnih dokumenata (eng. *Text document clustering - TDC*) predstavlja veoma važnu oblast naučnog istraživanja koja se rapidno razvija usled ogromne količine tekstualnih podataka proizvedenih sa strane Interneta, društvenih medija, tekstualnih poruka i drugih digitalnih izvora. Tehnike rudarenja teksta (eng. *text mining*) se koriste za obradu i analizu podataka zasnovanih na tekstualnim tipovima podataka. Upotreba različitih metoda rudarenja teksta omogućava izdvajanje relevantnih informacija, identifikaciju obrazaca, ključnih reči i drugih atributa iz tekstualnih podataka. Jedan od ključnih metoda u rudarenju teksta je klasterovanje, odnosno grupisanje teksta, koja ima za cilj da automatski podeli dokumente na konačan skup homogenih klastera (grupa). U određenom klasteru, svi dokumenti su međusobno slični na osnovu sadržaja, dok su u različitim klasterima sličnost je minimalan. Jedan od glavnih nedostataka tradicionalnih algoritama za klasterovanje je to što mogu lako da se zaglave u lokalnom optimumu [1]. Pokazalo se da su algoritmi inteligencije roja veoma efikasni u izbegavanju problema zaglavljivanja u lokalnom optimumu [2] i stoga su prikladniji za primenu klasterovanja.

Sistemi inteligencije roja generalno se sastoje od koloaborativne populacije jediniki (rešenja), koji lokalno sarađuju jedni sa drugima, kao i u interakciji sa svojim okruženjem. Svaki pojedinac u populaciji sledi jednostavna pravila, međutim individualno ponašanje nije inteligentno, već kolektivno ponašanje proizvodi inteligentno ponašanje koju vodi ka dostizanju zajedničkog cilja, odnosno pronalaska optimalnog ili blizu optimalnog rešenja u razumnom vremenskom periodu. Saradnja među pojedincima dovodi do evolucije inteligentnih globalnih performansi i ovo inteligentno kolektivno ponašanje se naziva inetligencija roja (eng. *Swarm intelligence - SI*).

Iz perspektive optimizacije, klasterovanje pripada klasi NP-teških problema optimizacije. Pokazalo se da su metaheuristički algoritmi veoma efikasni u rešavanje NP-teških problema optimizacije i rezultiraju bliskim optimalnim rešenjima u razumnom vremenskom periodu. Uopšte, metaheuristički algoritmi, a posebno algoritmi inteligencije roja, imaju mnogo uspešnih primena u različitim oblastima, kao što su segmentacija slika, klasifikacija, klasterovanje [3, 4, 5, 6, 7], selekcija atributa [8], grupisanje podataka [9], optimizacija arhitekture konvolucionih neuronskih mreža [10, 11, 12, 13, 14], bežične senzorske mreže [15, 16], aplikacije u kladu računarstvu

[17, 18, 19, 20, 21, 22], rutiranje vozila [23, 24], planiranje putanje [25], problemi pakovanja [26], alokacija resursa [27], optimizacija u različitim aplikacijama dubokog učenja i obuci neuronskih mreža [28, 29, 30, 31, 32], menadžment lanca snabdevanja [33] i mnogi drugi.

U ovoj disertaciji je predložen poboljšani hibridni algoritam inteligencije roja. Konkretno, algoritam optimizacije voćne mušice (eng. *Fruit-fly optimization - FFO*) [34, 35] koji je poboljšan i hibridizovan sa algoritmom svica (eng. *Firefly algorithm - FA*) [36] i dodatno u hibridni algoritam je ugrađen mehanizam učenja zasnovan na opoziciji (eng. *Opposition-based learning - OBL*). Za primenu rešavanje problema klasterovanja tekstualnih dokumenata, predložen poboljšani hibridni algoritam je kombinovan sa tradicionalnim algoritmom K-srednjih vrednosti (eng. *K-means*) [37]. Algoritam optimizacije voćne mušice je nedavno uspešno primenjen za grupisanje [38] ali ne i za specifičan problem grupisanja dokumenata i nije hibridizovan sa algoritmom svica. Poboljšani hibridni algoritam je nazvan kao hibridna poboljšana voćna mušica svitac (eng. *Hybrid enhanced fruit-fly firefly - HEFFF*).

## 1.1 Predmet i problem istraživanja

Klasterovanje tekstualnih dokumenata predstavlja značajnu oblast naučnog istraživanja koja se veoma brzo razvija zbog velike količine tekstualnih podataka. U ovoj oblasti istraživanja ima mnogo izazova, tekstualni podaci su predstavljeni u visokodimenzionalnom prostoru, varijacija u dužini tekstova takođe utiče na predstavljanje, potrebno je uzeti u obzir korelaciju između reči koje se pojavljuju u tekstualnim dokumentima kada se radi klasterovanje. Mnoge studije su koristile metaheurističke i nemetaheurističke pristupe za rešavanje problema klasterovanja tekstualnih dokumenata. Dobijeni rezultati postojećih studija su samo donekle uspešni u poboljšanju klasterovanja. Uprkos postojećim metaheurističkim aplikacijama za klasterovanje tekstualnih dokumenata i dalje postoji potreba da se razvije algoritam koji je robusniji, efikasniji od postojećih i koja prevazilazi nedostatke dosadašnjih tehnika. U TDC-u nemetaheuristički pristupi imaju problem sa inicijalizacijom početnih centroida klastera i takođe autlajeri u dokumentima predstavljaju problem, kao i zaglavljivanje u lokalnim optimumima, pored toga metaheuristički pristupi ne postižu uvek dobar balans između intenzifikacije i diverzifikacije u prostoru pretraživanja,

što znači da pronalazak ka globalnom optimumu nije zagarantovano.

Ovaj rad je inspirisan sa sledećim istraživačkim pitanjima:

- (i) Kako razviti efikasnu metodu za klasterovanje tekstualnih dokumenata tako što će unutar klastera biti dokumenti sa blisko povezanim temama, a sličnost sadržaja i tema će se razlikovati, odnosno minimizirati u različitim klasterima?
- (ii) Kako prevazilaziti nedostatke postojećih metoda kod problema klasterovanja tekstualnih dokumenata?
- (iii) Kako razviti efikasan algoritam kombinacijom metaheurističkih i nemetaheurističkih pristupa za klasterovanje tekstualnih dokumenata?

## 1.2 Cilj istraživanja

Klasterovanje tekstualnih dokumenata predstavlja važan zadatak, prema tome cilj ovog rada je razvoj metode za primenu problema klaseterizacije tekstualnih dokumenata, koja prevazilazi nedostatke postojećih metoda, kombinacijom metaheurističkih i nemetaheurističkih pristupa, kao i ugrađivanjem drugih mehanizama u algoritam.

U skladu sa navedenim ciljom, ova studija daje odgovore na prethodno definisana istraživačka pitanja na sledeći način:

- (i) Razviti poboljšanu metahurističku metodu u smislu kvaliteta rešenja da budu superiorne u odnosu na postojeće metode i kombinovati sa nemetaheurističkom tehnikom koja rezultira robusnijom i efikasnijom pristupom.
- (ii) Prevazilaziti problem inicijalizacije nemetaheurističkog pristupa prethodnim uvođenjem metaheurističke faze u proces algoritma, a u metaheuristički pristup kombinovati metode i ugraditi mehanizme koje prevazilazi probleme balansiranja eksploracije i eksploatacije.
- (iii) Koristiti novi hibridni pristup i dalje poboljšati sa tradicionalnim algoritmom K-srednjih vrednosti za klasterovanje tekstualnih dokumenata i proizvesti bolje rezultate u poređenju sa onima dobijenim drugim najsavremenijim pristupima za isti problem.

### 1.3 Hipoteze u istraživanju

Ovo istraživanje se bazira i usredsređuje na proveru i potvrđivanje sledećih hipoteza:

**Glavna hipoteza:** Primenom novih metoda za rešavanje problema klasterovanja tekstualnih dokumenata moguće je generisati bolje klastere u smislu jasnog razdvajanja klastera, u kojima su skupovi tekstualnih dokumenata znatno bliži po kontekstu jedan drugom nego oni koji nisu u istom klasteru.

**Posebne hipoteze:**

**Hipoteza 1 (H1):** *Kombinacija metaheurističkog i nemaetaheurističkog pristupa rezultira robusnom metodom za problem klasterovanja tekstualnih dokumenata.*

**Hipoteza 2 (H2):** *Hibridizacija komponenata različitih metaheurističkih metoda kao i ugrađivanje specifičnih mehanizama može imati značajan uticaj za poboljšanje kvaliteta rešenja i primenu konkretnog problema.*

**Hipoteza 3 (H3):** *Poboljšana hibridna metaheuristička metoda sa kombinacijom mašinskog učenja može da prevazilazi nedostatke postojećih tehnika i da bude efikasnija za rešavanje problema klasterovanja tekstualnih dokumenata.*

### 1.4 Metode istraživanja

U cilju uspešne realizacija rešavanje problema klasterovanja tekstualnih dokumenata i verifikacija postavljenih hipoteza, u ovoj disertaciji su korišćeni različiti metodi istraživanja. Analizom postojeće dostupne naučne literature se ukazuje na opravdanost predloženog modela za klasterovanje tekstualnih dokumenata. Metodom modelovanja je razvijem novi predloženi model koji je prvo testiran na neograničenim test funkcijama, a zatim na uobičajenim standardnim skupovima tekstualnih podataka. Statističkim metodama su potvrđena značajna poboljšanja metode i komparativnim metodom se upoređuje predloženi model sa postojećim modelima iz literature.

## 1.5 Naučni doprinos istraživanja

Naučni doprinos ove disertacije je:

1. Razvoj poboljšane hibridne metode HEFFF za globalnu optimizaciju.
2. Fuzija kvalitetnih komponeneta metaheuristike i OBL mehanizma koja rezultira kvalitetnim rešenjem.
3. Razvoj kombinovane metode metaheuristike HEFFF i mašinskog učenja K-srednjih vrednosti za rešavanje problema klasterovanja tekstualnih dokumenata koja rezultira optimalnim rešenjem.
4. Rešavanje problema klasterovanja tekstualnih dokumenata, razvijen model je upoređen sa drugim postojećim modelima i predložen model ima značajno poboljšanje u odnosu na postojeće tehnike.

Rezultati ove doktorske disertacije su objavljeni u prestižnom svetskom časopisu kategorizacije M21a. Kvalitet rada kao i postignuti rezultati su verifikovani sa strane eminentnih eksperta iz domena.

<sup>11</sup> **Timea Bezdan**, Catalin Stoean, Ahmed Al Naamany, Nebojsa Bacanin, Tarik A. Rashid, Miodrag Zivkovic, and K. Venkatachalam. "Hybrid fruit-fly optimization algorithm with k-means for text document clustering." *Mathematics*, Volume 9, Issue 16, MDPI, 2021, pp. 1929, [GoogleScholar] [CrossRef] [39].

## 1.6 Struktura rada

Uzimajući u obzir da problem ovog istraživanja pripada problemima optimizacije, nakon uvodnog dela, poglavlje 2 daje opis koncepta optimizacije kao i kategorizaciju optimizacionih problema. Zatim ova disertacija u poglavlju 3 daje opis mašinskog učenja, oblast kojoj klasterovanje pripada. Nakon toga u poglavlji 4 je dat sistematski pregled literature postojećih metoda koje su primenjene na problemu klasterovanje tekstualnih dokumenata.

Poglavlje 5 opisuje problem klasterovanja tekstualnih dokumenata, daje opis meta-heurističkih algoritama čiji komponenti su korišćeni u predloženoj hibridnoj metodi, takode ukazuje na njihove prednosti i nedostatke, i pored toga daje i opis specifičnih

mehanizama koji su ugrađeni u model, nakon toga sledi opis algoritma mašinskog učenja K-srednjih vrednosti koji se takođe koristi u predloženom modelu. Nakon opisa svih komponenata predstavljen je predloženi poboljšani hibridni metaheuristički metod, kao i procedura njene adaptacije na problem klasterovanja tekstualnih dokumenata.

U poglavlju 6 su prezentovani dobijeni rezultati simulacije na neograničenim test funkcijama, dobijeni rezultati eksperimenta klasterovanja tekstualnih dokumenata, kao i mere i statističke analize koje su korišćene u komparaciji sa drugim postojećim metodama.

Na kraju, poglavlje 7 je zaključak koji rezimira samo istraživanje i dobijene rezultate.

## 2 Optimizacija i kategorizacija optimizacionih problema

Optimizacija je svuda, većina procesa ima potencijal da se optimizuje. Optimizacija može da se pojavljuje kao maksimizacija ili minimizacija, na primer maksimizacija kvaliteta, profita, učinka, ili minimizacija troškova, vremena, i rizika. Većina stvarnih problema optimizacije u nauci, inženjerstvu, ekonomiji i biznisu su složeni, izazovni za rešavanje i pripadaju klasi NP-teških problema. Poslednjih godina, optimizacija za rešavanje složenih problema iz stvarnog sveta postaje sve interesantnija oblast istraživanja.

Egzaktne metode optimizacije su pogodne za rešavanje jednostavnih problema optimizacije, međutim kada je problem složeniji, prostor rešenja je veći, problem ima više lokalnih minimuma, tada tradicionalne metode često imaju loše performanse ili neuspeh u rešavanju problema optimizacije. Rešenja mogu biti zarobljeni u lokalnim optimumima i zbog toga algoritam postaje nesposoban da nastavi pretragu i ne garantira da će pronaći globalno najbolje rešenje. Mnogi problemi u stvarnom životu su složeniji i komplikovaniji, shodno tome, postoji potreba za korisnim, robusnim optimizatorom, stoga je interesovanje mnogih istraživača je privučeno da pronađu odličan pristup globalnoj optimizaciji. Korišćenje aproksimativnih algoritama je efikasno u rešavanju problema globalnih optimizacija. Oni ne garantuju najbolji globalni optimum, ali će se približiti globalnom optimumu i naći približno rešenje. Postoje dve vrste aproksimativnih algoritama, heuristički i metaheuristički algoritmi. U poslednje dve decenije razvijeni su i poboljšani brojni metaheuristički algoritmi inspirisani prirodom, koji su efikasni u rešavanju složenih problema optimizacije.

### 2.1 Koncept optimizacije i matematička formulacija

Optimizacija je proces pronalaženja najboljeg mogućeg rešenja za određeni problem, pod određenim skupom ograničenja. Praktično, sve probleme u realnom svetu možemo definisati kao zadatak optimizacije, stoga je optimizacija ključna paradigma u mnogim različitim domenima, kao što su planiranje, inženjerski dizajn, mreže, transport, proizvodnja, ekonomija, finansije i drugi.

Osnovni problem optimizacije se sastoji od tri komponente:

- varijabla za odlučivanje (jedna ili više ulaznih varijabli),
- funkcija cilja,
- ograničenja.

**Varijabla odlučivanja** Varijabla za odlučivanje mogu biti diskretna, kontinualna ili mešovita (diskretno-kontinualne) varijable.

**Funkcija cilja** Funkcija cilja  $f(k)$ , takođe nazvana kriterijumska funkcija ili funkcija troškova, odnosno gubitaka, je funkcija koju treba minimizirati ili maksimizirati zavisno od datog problema, drugim rečima, potrebno je pronaći najbolju vrednost za odlučivanje koja će minimizirati ili maksimizirati funkciju cilja.

**Ograničenja** Ograničenja (jednakosti ili nejednakosti) su funkcije koje treba da zadovolje varijable odlučivanja. Ako promenljiva odluke zadovoljava ograničenja, ona predstavlja izvodljivo rešenje (eng. *feasible solution*), inače se kaže da je neizvodljivo rešenje (eng. *infeasible solution*).

**Matematička formulacija** Problem optimizacije može se matematički definisati na sledeći način:

Data je varijabla za odlučivanje

$$x = (x_1, x_2, x_3, \dots, x_n) \quad (1)$$

koji minimizira funkciju cilja

$$f(x) \quad (2)$$

podložen ograničenjima

$$g_j \leq 0 \quad (j = 1, 2, \dots, J) \quad (3)$$

$$h_k = 0 \quad (j = 1, 2, \dots, K) \quad (4)$$

gde  $x_n$  je varijabla odlučivanja,  $f(x)$  predstavlja funkciju cilja,  $g_j$  i  $h_k$  su ograničenja nejednakosti i jednakosti, respektivno.

**Kategorizacija optimizacionih problema** Problemi optimizacije se mogu kategorisati na različite načine, na osnovu:

- funkcije cilja,
- prisustva ograničenja,
- prirode jednačina,
- odvojivosti funkcija,
- tipa varijabli odlučivanja,
- itd.

Kategorizacija zasnovana na funkciji cilja može se klasifikovati na jednociljne (eng. *single-objective*), višeciljne (eng. *multi-objective*) i probleme izvodljivosti (eng. *feasibility problem*). U jednociljnoj optimizaciji postoji samo jedna funkcija cilja, dok problemi sa višestrukim ciljevima imaju više od jedne funkcije cilja. Problemi izvodljivosti su ekstremni slučajevi kada postoje samo ograničenja, a ciljne funkcije uopšte ne postoje, u ovim problemima svako izvodljivo rešenje predstavlja optimalno rešenje.

Problemi optimizacije bazirani na ograničenjima mogu biti ograničeni ili neograničeni.

Priroda jednačina kod optimizacionih problema mogu se klasifikovati kao linearna, nelinearna, geometrijska i kvadratna. Ako su sve funkcije, ciljevi i ograničenja su linearni, problem je linearnog programiranja (eng. *Linear programming - LP*), ukoliko bilo koja funkcija je nelinearna, tada je problem nelinearnog programiranja (eng. *Nonlinear programming problem - NLP*), kada su funkcije izražene polinomima, to je problem geometrijskog programiranja (eng. *Geometric programming problem - GMP*), problem kvadrnog programiranja (eng. *Quadratic programming - QP*) ima kvadratnu funkciju cilja i linearna ograničenja.

Na osnovu odvojivosti, optimizacioni problemi mogu se klasifikovati na odvojive i neodvojive. Odvojive optimizacione probleme je lakše rešiti jer su promenljive funkcije zavisne od drugih promenljivih, za razliku od neodvojivih problema, gde su promenljive funkcije nezavisne od drugih promenljivih [40].

Na osnovu varijabli odlučivanja, optimizacioni problemi mogu se klasifikovati na diskretne (celobrojne), kontinuirane (stvarne vrednosti) i mešovite varijable (diskretne i kontinuirane). Problemi koji uključuju diskretne varijable, nazivaju se diskretnom optimizacijom ili kombinatornom optimizacijom. Kombinatorna optimizacija je opisana kao metod pronalaženja optimalne organizacije, uređenja, grupisanja ili izbora diskretnog, ograničenog broja mogućih rešenja. Diskretna optimizacije dalje može biti klasifikovana na probleme celobrojnog programiranja (eng. *Integer programming - IP*), mešovito celobrojno <sup>26</sup> linearno programiranje (eng. *Mixed-integer linear programming - MILP*) i mešovito celobrojno <sup>40</sup> nelinearno programiranje (eng. *Mixed-integer nonlinear programming - MINLP*). Neki od poznatih slučajeva problema kombinatorne optimizacije su problem trgovačkog putnika (eng. *Traveling salesman problem - TSP*), <sup>40</sup> problem rutiranja vozila, problemi sa pakovanjem, alokacija resursa, itd. Problemi koji uključuju kontinuirane varijable se nazivaju kontinuirana optimizacija ili globalna optimizacija, koja se može klasifikovati na ograničene i neograničene probleme.

Nakon što se definiše problem optimizacije, potreban je adekvatan algoritam optimizacije za pronalaženje rešenja za dati problem. Elementarna klasifikacija algoritama optimizacije je deterministička, nedeterministička (stohastička) ili hibridna. Deterministički algoritmi uvek daju isti izlaz, sa druge strane, nedeterministički algoritmi uključuju slučajnost, isti unos će vratiti različite rezultate pri različitom izvršavanju.

## 2.2 Računska složenost

Računska složenost (eng. *computational complexity*) se koristi za merenje koliko je lako ili teško rešiti problem optimizacije pomoću određenog algoritma. Klasifikacija obuhvata klasu složenosti P, odnosno složenost determinističkog polinomskog vremena (eng. *deterministic polynomial-time complexity*) i klasu složenosti NP, to jest složenost nedeterminističkog polinomskog vremena (eng. *non-deterministic polynomial-time complexity*).

Problemi koji su rešeni u determinističkom polinomskom vremenu osdnose se na klasu P, odnosno klasu problema koji se mogu rešiti (laki problemi).

Problemi u NP su skup problema odlučivanja gde je odgovor ili da ili ne („da-ne” problemi) i koji se mogu rešiti u nedeterminističkom polinomskom vremenu, a rešenje

se može proveriti u polinomskom vremenu. Problemi koji su NP-teški pripadaju klasi zadataka koji su barem jednako teški kao i najteži problemi u NP. Neki od primera NP-teških problema su problemi sekvenciranja rasporeda, problemi sa rutiranjem, itd. NP-kompletan je klasa nerešivih (teških) problema.

## 2.3 Algoritmi stohastičke aproksimacije

Većina problema optimizacije u stvarnom svetu su teško rešive, pripadaju NP-teškoj klasi, nelinearni, multimodalni, složeni, prostor pretrage je veliki, izračunavanje optimalnog rešenja je nemoguće u razumnom vremenu i ne može se rešiti tačnim tehnikama optimizacije. Većina tradicionalnih algoritama je prikladno za lokalnu pretragu i nisu prikladni za globalnu optimizaciju. Za ove vrese probleme koriste se aproksimativni algoritmi koji se mogu podeliti na heurističke i metaheurističke. Heuristički pristupi više zavise od problema, s druge strane, metaheuristika je prikladna za široku primenu seriju problema optimizacije. Ovi algoritmi ne obezbeđuju najbolje rešenje, već pružaju zadovoljavajuće rešenje koje je blizu optimalnom rešenju i mogu da se pronađu u razumnom vremenu.

## 2.4 Metaheuristički algoritmi

Glove [41], 1986. godine je prvi put koristio termin metaheuristika, koji je kombinacija dve grčke reči "*meta*" i "*heuristika*". Meta na grčkom znači izvan, metoda višeg nivoa, dok heuristika znači istraživanje, otkrivanje strategija za rešavanje problema.

Metaheuristički algoritmi su stohastički algoritmi, koje karakterišu slučajnost. Ovi algoritmi imaju dve glavne faze pretrage, eksploracija (diverzifikacija) i eksploatacija (intenzifikacija). U fazi eksploracije, algoritam otkriva prostor pretrage na globalnom nivou kreiranjem različitih rešenja. U fazi eksploatacije, algoritam konvergira na lokalnom regionu blizu najboljih rešenja i biraju se najbolji kandidati rešenja. Metaheuristički algoritmi su pogodni za rešavanje velikih, teških optimizacionih problema i njihovo dizajniranje i implementacija je relativno jednostavna.

Metaheuristički algoritmi inspirisani prirodom mogu da se podele u dve glavne kategorije: evolutivni algoritmi (eng. *Evolutionary algorithm* - *EA*) i algoritmi inteligencije roja (eng. *Swarm intelligence* - *SI*).

## 2.5 Evolutivni algoritmi

Evolutivni algoritmi su inspirisani procesom biološke evolucije, i mogu se podeliti u tri kategorije:

- evoluciono programiranje (eng. *evolutionary programming*) [42],
- evolucione strategije (eng. *evolutionary strategies*) [43],
- genetski algoritmi (eng. *genetic algorithms*) [44].

Glavni elementi dizajna evolutivnih algoritama su:

- inicijalizacija populacije,
- evaulacija funkcija cilja,
- strategija selekcije,
- strategija reprodukcije,
- strategija zamene,
- kriterijumi za zaustavljanje.

Najčešće korišćene strategije su *roulette wheel* selekcija, *tournament* selekcija, stohastičko univerzalno uzorkovanje i selekcija zasnovana na rangu. U fazi reprodukcije od starih se formiraju nove jedinke, i to se postiže korišćenjem operatora varijacije kao što su mutacija (operator unarne varijacije) i ukrštanje (operator binarne varijacije).

Operator mutacije se primenjuje na jednu jedinku kako bi se napravile male modifikacije izabrane jedinke. Najčešće korišćeni operatori mutacije su bitflip mutacija, slučajna mutacija, delta mutacija, Gausova mutacija. Operator ukrštanja kombinuje informacije o jedinkama dva roditelja da bi proizveo novu jedinku. Poznati operatori ukrštanja su ukrštanje sa jednom tačkom, ukrštanje sa više tačaka, uniformno ukrštanje. Strategija zamene se odnosi na selekciju preživljavanja, gde treba da se napravi izbor koji će pojedinci biti odabrani za sledeću generaciju.

Pseudokod opšteg procesa evolutivnih algoritama je predstavljen u algoritmu 1.

Dijagram toka opšteg dizajna evolutivnih algoritama je prikazan na slici 1.

---

**Algoritam 1** Opšti pseudokod evolutivnih algoritama

---

Inicijalizacija populacije

Evaulacija rešenja

**while** Nije zadovoljen kriterijum zaustavljanja **do**

    Ukrštanje

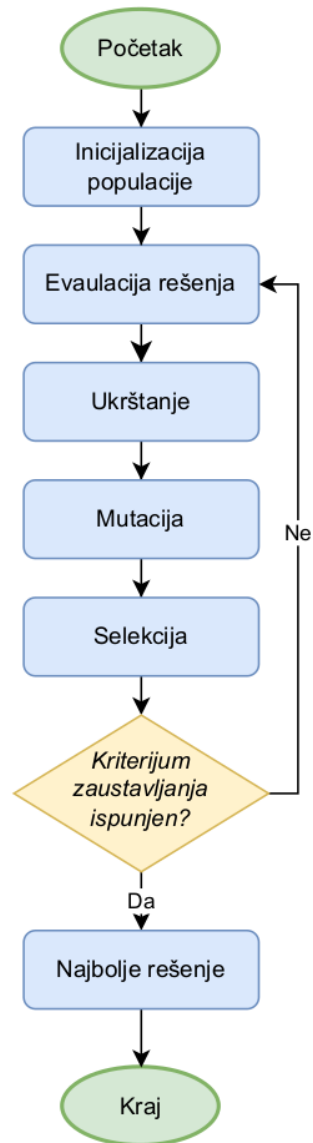
    Mutacija

    Evaulacija rešenja i selekcija

**end while**

Memorisanje najboljeg rešenja

---



Slika 1: Dijagram toka opšteg dizajna evolutivnih algoritama

## 2.6 Algoritmi inteligencije roja

Algoritmi inteligencije roja su inspirisani kolektivnom inteligencijom decentralizovanog, samoorganizovanog sistema. Grupna inteligencija se razvila iz prirode, na primer od modela košnice, uzgoj životinja, kolonije mrava, rast bakterija, jato ptica, itd. Ponašanje pojedinaca nije inteligentno, međutim celokupna populacija vodi ka inteligentnoj globalnoj postupku. Grupno ponašanje proizlazi iz interakcije među pojedincima na lokalnom nivou, kao i sa njihovom okolinom.

Izvestan broj algoritama zasnovanih na roju inteligencije je razvijen različitim dizajnom, ali svi oni poseduju nekoliko opštih svojstava. Prvenstveno, populacija se nasumično inicijalizuje u definisanom prostoru pretrage, a vektor rešenja predstavlja svakog pojedinca u populaciji. Randomizacija se koristi radi istraživanja područja pretraživanja i izbegavanja zaglavljivanja u lokalnom optimumu. Nakon svake iteracije, lokacija pojedinaca se pomera, a na osnovu vrednosti ciljne funkcije jedinki, bira se najbolje rešenje za sledeću generaciju, odnosno sledeću iteraciju u algoritmu. Na ovaj način se sistem evoluira, a takav pristup selekcije omogućava konvergenciju do globalnog (ili blizu globalnog) optimuma.

Pseudokod opšteg procesa algoritma inteligencije roja je predstavljen u algoritmu 2.

---

**Algoritam 2** Opšti pseudokod algoritma inteligencije roja

---

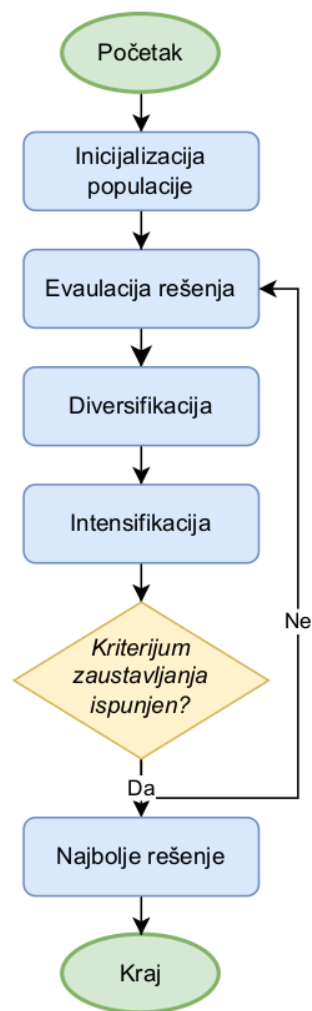
```
Inicijalizacija populacije
Evaluacija rešenja
while Nije zadovoljen kriterijum zaustavljanja do
    Diverzifikacija
    Intenzifikacija
    Evaluacija rešenja
end while
Memorisanje najboljeg rešenja
```

---

Dijagram toka opšteg dizajna algoritama inteligencije roja je prikazan na slici 2.

## 2.7 Hibridni metaheuristički algoritmi

Hibridizacija metaheurističkih algoritama je veoma popularna kategorija metaheurističke studije. <sup>4</sup> Hibridni algoritam je paralelna ili distribuirana implementacija dva ili više algoritama, koji kombinuje prednosti različitih metaheurističkih pristupa



Slika 2: Dijagram toka opšteg dizajna algoritama inteligencije roja <sup>4</sup> algoritama

koristeći algoritamske komponente iz različitih metoda optimizacije ili mešavinu metaheurističkih metoda sa različitim metodama veštačke inteligencije. Tehnika hibridizacije rezultira sinergističkom sintezom spojenih algoritama ili komponenata <sup>4</sup> algoritama. Neki metaheuristički algoritmi imaju brzu konvergenciju, dok drugi imaju sporu konvergenciju. Hibridni optimizatori su često znatno moćniji u pogledu stope konvergencije i kvaliteta rešenja, oni mogu proizvesti efikasnije performanse i veću fleksibilnost kada se bave složenim realnim problemima iz stvarnog sveta.

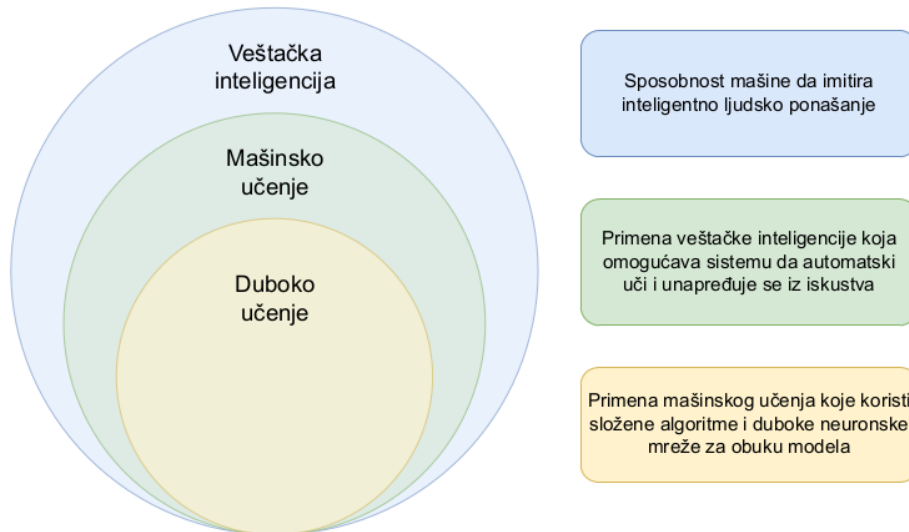
Suštinski deo dizajna metaheurističkog algoritama je da se napravi optimalna ravnoteža globalne diverzifikacije i lokalne intenzifikacije. <sup>4</sup> Rana konvergencija je posledica male raznovrsnosti, s druge strane, veća raznolikost olakšava pažljivo istra-

živanje, omogućava bolje istraživanje prostora pretrage i izbegava zaglavljivanje u lokalnom optimumu. <sup>4</sup> Ako je algoritam bolje u eksploataciji, ali loš u eksploraciji, slab je u skoku <sup>4</sup> sa lokalnog optimuma, zbog toga imaju manju verovatnoću da dođe do globalnog optimuma, ukoliko <sup>4</sup> dati problem ima mnogo lokalnih optimuma. U slučaju da uspe da napusti lokalni optimum, može dobro da funkcioniše pri približavanju globalnom optimumu. Shodno tome, ključno je imati pravu ravnotežu u procesu pretraživanja.

U procesu hibridizacije algoritam postaje složeniji, stoga je <sup>4</sup> potrebno strukturu sastaviti tako da bude jednostavna za implementaciju.

### 3 Mašinsko učenje

Mašinsko učenje (eng. *Machine learning* - *ML*) je grana veštačke inteligencije (eng. *Artificial intelligence* - *AI*) i računarske nauke koja korišćenjem algoritama uči iz podataka i postepeno poboljšava tačnost predikcije. Cilj mašinskog učenja je da predvidi buduće događaje ili scenarija koja su nepoznata računaru. Slika 3 ilustruje odnos između veštačke inteligencije, mašinskog učenja i dubokog učenja.



Slika 3: Veštačka inteligencija, mašinsko učenje i duboko učenje

Mašinsko učenje postaje sve rasprostranjeniji i ima brojne oblasti primene, kao što je kategorizacija teksta, prevođenje teksta, prepoznavanje slika, prepoznavanje lica, prepoznavanje govora, analiza društvenih mreža, prepoznavanje objekta, primena u medicini, sportu, finansiji, bankarstvu i mnogim drugim oblastima.

Artur Semjuel, jedan od pionira veštačke inteligencije, je kreirao pojam mašinsko učenje 1950-tih godina i definisao ju je kao oblast proučavanja koja računarima daje mogućnost da uče bez eksplicitnog programiranja.

Godine 1950. matematičar Alan Turing u svom radu "Računarska mašina i inteligencija" je uveo standardnu meru za testiranje mašinske inteligencije i iz ovog rada je proizašao Turingov test, prema čemu mašina mora da bude inteligentna i da

reaguje na način koji se ne može razlikovati od ljudskog bića. Ukoliko ne može da se napravi razlika između računara i ljudi, onda se kaže da je računar prošao Turingov test.

Tom M. Mitchell (1998) je definisao mašinsko učenje na sledeći način: "Ža kompjuterski program se kaže da uči iz iskustva  $E$  (eng. *experience*) u odnosu na neku klasu zadataka  $T$  (eng. *task*) i meru učinka  $P$  (eng. *performance*), ako se njegov učinak na zadacima  $T$ , mereno sa  $P$ , poboljšava se sa iskustvom  $E$ ."

Sistem učenja algoritma mašinskog učenja se deli na tri glavna dela: proces odlučivanja, funkcija greške i proces optimizacije.

**Proces odlučivanja** Generalno, algoritmi mašinskog učenja se koriste za predviđanje ili klasifikaciju. Na osnovu ulaznih podataka, koji mogu biti označeni ili neoznačeni, a algoritam će proizvesti procenu o obrascu u podacima.

**Funkcija greške** Funkcija greške procenjuje predviđanje modela. Ako postoje poznati primeri, funkcija greške može da napravi poređenje da proceni tačnost modela.

**Proces optimizacije modela** Ako model može bolje da odgovara tačkama podataka u skupu za obuku, onda se težine prilagođavaju da bi se smanjila neslaganja između poznatog primera i procene modela. Algoritam će ponoviti ovaj proces "procene i optimizacije", samostalno ažurirajući težine dok se ne dostigne prag tačnosti.

Taksonomija tehnike mašinskog učenja na osnovu zadataka:

1. Nadgledano učenje (eng. *Supervised learning*)
2. Nenadgledano učenje (eng. *Unsupervised learning*)
3. Polunadgledano učenje (eng. *Semi-supervised learning*)
4. Učenje sa pojačanjem (eng. *Reinforcement learning*)

### 3.1 Nadgledano učenje

Nadgledano učenje (eng. *supervised learning*) koristi označeni skup podataka za obuku algoritama da klasifikuju podatke ili tačno predvide rezultate. Označeni

skupovi podataka su oni, gde se zna ciljni izlaz, dakle kod nadgledanog učenja, ulazni podaci za obuku modela dolaze u paru  $(X, y)$ , gde je  $X$  nezavisna varijabla sa svim karakteristikama ulazne instance, a  $y$  je zavisna varijabla (ciljna varijabla ili oznaka) koja predstavlja odgovarajuću oznaku ulazne instance, to jest predmet predviđanja. Cilj obučavanja modela je minimizacija grešaka za dati skup podataka i generalizacija za nove, nepoznate ulazne podatke.

Na osnovu vrste problema, nadgledano učenje može da se podeli na probleme klasifikacije i probleme regresije. Kod klasifikacionih problema, ciljna varijabla je kategorička (celobrojna), dok kod regresionih problema se predviđaju kontinualne vrednosti.

### 3.2 Nenadgledano učenje

<sup>52</sup> Nenadgledano učenje (eng. *unsupervised learning*) je drugi najčešći oblik mašinskog učenja. Kod nenadgledanog učenja, algoritmi sprovode obuku na neoznačenim skupovima podataka, prema tome, željeni rezultat je nepoznat, a cilj algoritama je otkrivanje skrivenih struktura, odnosa i obrazaca u podacima. Ovaj pristup je suprotan od nadgledanog učenja, gde se modeli uče na označenim podacima, kao što je prethodno opisano.

Neki od najčešće korišćenih pristupa nenadgledanog učenja uključuju:

- klasterovanje (eng. *clustering*),
- smanjenje dimenzionalnosti (eng. *dimensionality reduction*),
- otkrivanje anomalija (eng. *anomaly detection*),
- učenje modela latentne promenljive (eng. *learning latent variable models*)

Procesom redukcije dimenzionalnosti (eng. *dimensionality reduction*) se smanji broj karateristika (eng. *features*) ulazih podataka. Dva uobičajena pristupa za smanjenje dimenzije su analiza glavnih komponenti (eng. *Principal component analysis - PCA*) i dekompozicija singularnih vrednosti (eng. *Singular value decomposition - SVD*).

### 3.2.1 Klasterovanje

Klasterovanje se odnosi na partitionisanje skupa ulaznih podataka na fiksni broj blisko povezanih podskupova, drugim rečima, grupisanja podataka u klustere. Kod klasterovanje cilj je pronalaženje „prirodnih grupa” među objektima, gde objekti unutar jedne grupe imaju visoku sličnost, a sličnost među različitim grupa je niska.

Problem klasterovanje može da se formuliše sledećom definicijom:

**Definicija** Dat je skup podataka  $D = \{v_1, v_2, \dots, v_n\}$  i celobrojna vrednost za broj klastera  $k$ , problem klasterovanje je da se definiše mapiranje  $f : D \rightarrow \{1, \dots, k\}$ , gde je svakom  $v_i$  je dodeljen jedan klaster  $C_j$ ,  $1 \leq j \leq k$ . Klaster  $C_j$  sadrži one vektore podataka koji su precizno mapirani u njega, odnosno  $C_j = \{v_i | f(v_i) = C_j, 1 \leq i \leq n, v_i \in D\}$ .

Jedna instanca, na primer  $v_1$  predstavlja vektor obeležja  $x$ , definisano kao  $v_1 = \{x_1, x_2, \dots, x_n\}$ .

Prema [1], tradicionalni algoritmi za klasterovanje mogu se grupisati u 9 klasa, kao što je ilustrovano u tabeli 1.

**Tabela 1:** Kategorije tradicionalnih algoritama za klasterovanje i primeri svakog tipa

| Kategorija                        | Primer                                                           |
|-----------------------------------|------------------------------------------------------------------|
| Algoritmi zasnovanih na particiji | K-means [37], K-medoids [45], PAM [46], CLARA [46], CLARANS [47] |
| Hijerarhijski algoritmi           | BIRCH [48], CURE [49], ROCK [50], Chameleon [51]                 |
| Algoritmi fuzzy teorije           | FCM [52], FCS [53], MM [54]                                      |
| Distribicioni algoritmi           | DBCLASD [55], GMM [56]                                           |
| Algoritmi gustine                 | DBSCAN [57], OPTICS [58], Mean-shift [59]                        |
| Algoritmi graf teorije            | CLICK [60], MST [61]                                             |
| Grid algoritmi                    | STING [62], CLIQUE [63]                                          |
| Algoritmi fraktalne teorije       | FC [64]                                                          |
| Algoritmi bazirani na modelima    | COBWEB [65]                                                      |

### 3.2.2 Mere sličnosti koje se koriste kod klasterovanja

Kvalitet rezultata klasterovanja zavisi od mere sličnosti koju koristi metoda i njene implementacije. Obično, tačka podataka koje treba grupisati su u visokodimenzionalnom prostoru i sličnost među njima se definiše pomoću mere udaljenosti ili sličnosti, kao što su:

- Euklidska udaljenost,
- Minkovski udaljenost
- Menhetn udaljenost,
- Srednja udaljenost,
- Ponderisana Euklidska udaljenost,
- Udaljenost tetive,
- Mahalanobisova udaljenost,
- Kosinusna sličnost,
- Pirsonova korelacija.

**Minkovski udaljenost** Minkovski udaljenost je definisana kao sledeće:

$$d_{min} = \left( \sum_{i=1}^n |x_i - y_i|^m \right)^{\frac{1}{m}} \quad (5)$$

gde  $m \geq 1$ ,  $x_i$  i  $y_i$  su dva vektora u  $n$ -dimenzionalnom prostoru.

**Euklidska udaljenost** Jedan od najpoznatijih mera udaljenosti koji se koristi za numeričke podatke je Euklidska udaljenost, koja predstavlja poseban slučaj Minkovske udaljenosti, kada je  $m = 2$ , mera je definisana kao sledeće:

$$d_{min} = \left( \sum_{i=1}^n |x_i - y_i|^2 \right)^{\frac{1}{2}} \quad (6)$$

**Menhetn udaljenost** Menhetn udaljenost je specijalni slučaj Minkovske udaljenosti, kada je  $m = 1$ , mera je definisana kao sledeće:

$$d_{min} = \sum_{i=1}^n |x_i - y_i| \quad (7)$$

**Srednja udaljenost** Srednja udaljenost je definisana sledećim izrazom:

$$d_{min} = \left( \frac{1}{n} \sum_{i=1}^n (x_i - y_i)^2 \right)^{\frac{1}{2}} \quad (8)$$

**Ponderisana Euklidska udaljenost** Ponderisana Euklidska udaljenost je modifikovana verzija Euklidske distance i definisana je kao:

$$d_{min} = \left( \sum_{i=1}^n w_i (x_i - y_i)^2 \right)^{\frac{1}{2}} \quad (9)$$

**Udaljenost tetive** Udaljenost tetive je takođe jedna modifikacija Euklidskog rastojanja. Udaljenost tetive je formulisana kao:

$$d_{min} = \left( 2 - 2 \frac{\sum_{i=1}^n x_i y_i}{\|x\|_2 \|y\|_2} \right)^{\frac{1}{2}} \quad (10)$$

gde  $\|x\|_2$  predstavlja  $L^2$ -normu  $\|x\|_2 = \sqrt{\sum_{i=1}^n x_i^2}$ .

**Mahalanobisova udaljenost**

**Kosinusna sličnost** Kosinusna sličnost između dva podataka se računa sledećom formulom:

$$\cos(\theta) = \frac{\sum_{i=1}^n x_i y_i}{\|x\|_2 \|y\|_2} \quad (11)$$

gde  $\|y\|_2$  je Euklidsku norma vektora  $y$  i definisana kao  $\|y\|_2 = \sqrt{y_1^2 + y_2^2 + \dots + y_n^2}$ .

**Pirsonova korelacija**

$$r = \frac{\sum_{i=1}^n (x_i - \mu_x)(y_i - \mu_y)}{\sqrt{\sum_{i=1}^n (x_i - \mu_x)^2} \sqrt{\sum_{i=1}^n (y_i - \mu_y)^2}} \quad (12)$$

gde  $\mu_x$  i  $\mu_y$  su prosečne vrednosti za  $x$  i  $y$ .

### 3.3 Polunadgledano učenje

<sup>62</sup> Polunadgledano učenje predstavlja kombinaciju nadgledanog i nenadgledanog učenja. Prilikom obuke, kod polunadgledanog učenja, koristi se manji skup označenih podataka i veći skup neoznačenih podataka za generisanje modela. Kada je u pitanju ogroman set podataka, označavanje je dugotrajan proces i iziskuje kvalifikovanu ljudsku radnu snagu što može biti skupo i nekada neizvodljivo. Sa druge strane, neoznačeni podaci su relativno jeftini i dostupni. Zbog navedenih razloga, polunadgledano učenje predstavlja dobro rešenje za kreiranje modela kada na raspolaganju stoji manji broj označenih podataka i veliki broj neoznačenih podataka. Pomoću polunadgledanog učenja, inicijalni model se trenira na malom broju označenih uzoraka, zatim se primenjuje na neoznačeni set podataka i dodeljuju se pseudo oznake, zatim se poboljšava iterativnim postupkom.

### 3.4 Učenje sa pojačanjem

Učenje sa pojačanjem (eng. *Reinforcement learning* - *RL*) obučava modele putem pokušaja i grešaka u cilju preduzimanja najbolje akcije tako što se nagrada maksimizuje u određenoj situaciji. Kod pojačanog učenja, agenti za pojačavanje odlučuju šta da se uradi za izvršavanje datog zadatka i agent se uči na osnovu iskustva. Radi se o učenju optimalnog ponašanja u određenom okruženju i donošenje najbolje odluke koja će maksimizirati nagradu.

## 4 Pregled literature klasterovanja tekstualnih dokumenata

U ovom odeljku su prezentovani najpopularniji metaheuristički algoritmi koji su korišćeni za klasterovanje tekstualnih dokumenata.

### 4.1 Klasterovanje tekstualnih dokumenata algoritmom optimizacije roja čestica

Studija [66] je baziran na tehnici optimizacije roja čestica (eng. <sup>9</sup> *Particle swarm optimization - PSO*) [67]. *Hasanzadeh et al.* su kombinovali PSO sa LSI. LSI je korišćen za smanjenje dimenzije tekstualnih dokumenata, dok PSO je korišćen za klasterovanje tekstualnih podataka. Za bolji balans između diverzifikacije i intenzifikacije, PSO je modifikovan sa prilagodljivom inercionom težinom. Nakon toga, PSO je kombinovan sa LSI za efikasno grupisanje korpusa. Rezultati rada pokazuju da predložena tehnika nadmašuje tehniku PSO sa K-srednjih vrednosti.

Karol i Manga [68] su predstavili dve hibridne metode koje su bazirani na algoritmu PSO za klasterovanje tekstualnih dokumenata, naime KPSO i FCPSO, ove metode kombinuju PSO sa tradicionalnim algoritmom K-srednjih vrednosti i *Fuzzy C-means* algoritam. K-srednjih vrednosti se koristi u KPSO-u za inicijalizaciju PSO algoritma za efikasno klasterovanje tekstualnih dokumenata. Slično kao kod KPSO, i FCM se koristi za generisanje početnih klastera u FCPSO, zatim PSO se koristi za generisanje najboljih optimalnih klastera. Predložene metode su testirani na *Reuters, 20 Newsgroups* skupovima podataka i upoređene su sa standardnim algoritmom K-srednjih vrednosti i FCM algoritmom.

Procena performanse metode PSO, K-srednjih vrednosti, kao i hibridne metode PSO sa algoritmom K-srednjih vrednosti za klasterovanje tekstualnih dokumenata je predstavljen u [69]. Simulacije su izvedene na tekstualnim skupovima podataka koje su napisane na nepalskom jeziku. Rezultati studije pokazuju da hibridna metoda PSO sa algoritmom K-srednjih vrednosti je nadmašio pojedinačne algoritme PSO i K-srednjih vrednosti.

Aghdam i Heidari [70] su kombinovali PSO i algoritam K-srednjih vrednosti za klasterovanje tekstualnih dokumenata. Autori su koristili PSO za selekciju najkori-

snijih i relevantnih atributa, to jest reči u vokabularu, za smanjenje broja ulaznih varijabli u model za klasterovanje tekstualnih dokumenata. Izabrani podskup ulaznih atributa se koristi kao ulaz u algoritam K-srednjih vrednosti. Predloženi metod je testiran na *Reuters* skupu podataka i upoređen sa metodama GA, IG i CHI. Rezultati potvrđuju efikasnost PSO-a za odabir najkorisnijih atributa za predstavljanje tekstualnih dokumenata pre primene algoritma K-srednjih vrednosti.

Song et al. [71] su predložili hibridni evolucioni računarski pristup, kombinovani PSO sa kvantnim ponašanjem (eng. *Quantum-behaved particle swarm optimization* - *QPSO*) i genetskim algoritmom (eng. *Genetic algorithm* - *GA*) [72] za rešavanje problema klasterovanja tekstualnih dokumenata. U studiji, autori su koristili robusnu sposobnost GA-a za globalno pretraživanje da bi poboljšali inicijalizaciju čestica u QPSO. Takođe, u studiji je predložena nova strategija pretrage za normalizaciju regiona. Simulacije su izvođene na skupovima podataka *20 Newsgroup* i *Reuters* u cilju testiranja efikasnosti predložene metode. Rezultati studije pokazuju da predloženi pristup nadmašio originalni GA, QPSO i poboljšani QPSO u smislu F-mere i ciljne funkcije.

Bharti i Singh [73] su predložili hibridni pristup selekcije atributa za klasterovanje tekstualnih dokumenata. Predloženi pristup kombinuje binarni PSO sa haotičnom mapom, učenjem zasnovanim na opoziciji, mutacijom i dinamičkom težinom inercije zasnovanom na kondiciji. Da bi se proverila efikasnost predloženog metoda, simulacije su sprovedene na tri standardnih skupova tekstualnih podataka *WebKB*, *Classic 4* i *Reuters*. Dobijeni rezultati eksperimenta pokazuju da predloženi pristup pronalazi prikladniji podskup atributa od komparativnih metoda TV, SGA, CBPSO i AIWPSO. Pored efikasne selekcije atributa, predloženi petod poboljšava brzinu konvergencije algoritma BPSO.

Abualigah et al. [74] su predložili metod FSPSOTC, koji se bazira na algoritmu PSO za odabir najkorisnijeg atributa za predstavljanje tekstualnih dokumenata u klasterovanju teksta. Izabrani podskup atributa se koristi kao ulaz u algoritam K-srednjih vrednosti. Predloženi metod je testiran na *Reuters* i *20 Newsgroups* skupovima podataka. Rezultati pokazuju da korišćenje PSO metode za odabir atributa pre primene algoritma K-srednjih vrednosti znatno poboljšavaju performanse u odnosu na druge metode selekcije atributa.

Daoud et al. [75] su kombinovali PSO pristup sa klasičnim algoritmom K-srednjih vrednosti da bi rešili problem inicijalne selekcije centroida pomoću K-srednjih vrednosti u grupisanju arapskog teksta. Predloženi metod je testiran korišćenjem tekstualnih podataka sa tri različita veb stranica: *BBC*, *CNN* i *OSAC*. Eksperimentalni rezultati pokazuju da je predloženi pristup nadmašio klasični algoritam K-srednjih vrednosti u klasifikaciji arapskog teksta u sva tri tekstualna skupa podataka u smislu preciznosti, odziva, F-mere i tačnosti.

Kushwaha i Pant [76] su koristili PSO koji je zasnovan na linku (eng. *Link based particle swarm optimization - LBPSO*) za izbor atributa u klasterovanju tekstualnih dokumenata. U LBPSO su ugrađeni operatori ukrštanja i mutacije iz genetskog algoritma radi poboljšanja stope konvergencije. Za odabir najinformativnijih atributa, LBPSO koristi novi mehanizam odabira suseda. Ovi informativni atributi se zatim procenjuju korišćenjem K-srednjih vrednosti da bi se poboljšala efikasnost algoritma i smanjila njegova vremenska složenost. LBPSO poboljšava sposobnost eksploracije i eksploatacije pomoću ažuriranja pozicije na osnovu najboljih pozicija susednih rešenja. Eksperimentalni rezultati korišćenjem skupova podataka *TDT2*, *Reuters*, i *Tr11* pokazuju da je LBPSO superiorni od druga dva komparabilna pristupa, BPSO i HFSPSOTC u pogledu odabira najinformativnijih atributa iz ovih skupova podataka.

Janani i Vijayarani [77] su predložili pristup klasterovanja tekstualnih dokumenata koji kombinuje PSO sa spektralnim klasterisanjem (eng. *Spectral Clustering Particle Swarm Optimization - SCPSO*). SCPSO pristup je testiran na standardnim skupovima podataka *Reuters*, *20 Newsgroups* i *TDT2*. Dobijeni rezultati su upoređeni sa tradicionalnim PSO, metodom očekivane maksimizacije (eng. *Expected maximization method - EMM*) i tehnikama klasterisanja sfernih K-srednjih vrednosti. U pogledu tačnosti klasterovanja, rezultati simulacije pokazuju da je SCPSO nadmašuje druge tehnike.

## 4.2 Klasterovanje tekstualnih dokumenata genetskim algoritmom

46

genetic algorithm method based on a latent semantic model (GAL)

Song i Park [78] su koristili poboljšani genetski algoritam baziran na *latent semantic* (LS) modelu, za problem klastrizacije tekstualnih dokumenata. LS model

je korišćen umesto vektorskog modela prostora za reprezentaciju tekstova. U predloženom pristupu, poboljšani GA je zasnovan na promenljivoj dužini stringa, a LS pomaže u istraživanju latentne semantike koju implicira upit kroz reprezentaciju u redukovanom domenionalnom prostoru. Rezultati simulacije su dobijeni korišćenjem standardnog skupa podataka *Reuters* i potvrđuju efikasnost i superiornost klasterovanja u odnosu na tradicionalni GA.

Premalatha i Natarajan [79] su kreirali metod za klasterovanje tekstualnih dokumenata koji je zasnovan na genetskom algoritmu i uključuje rangiranu stopu mutacije kao i operator simultane mutacije. U simultanoj mutaciji, GA je koristio više operatora mutacije u isto vreme da napravi sledeću generaciju. Odnos mutacije operatora varira u odnosu na vrednost potomstva koje pravi. Stopa mutacije na hromozomu u rangiranoj strategiji se prilagođava na osnovu ranga ciljne funkcije prethodne populacije. Rezultati simulacije su upoređeni sa skupovima podataka *CISI*, *ADI* i *Cranfield*. Rezultati studije pokazuju da predloženi metod nadmašuje tradicionalni algoritam K-srednjih vrednosti i tradicionalni GA u pogledu statističkih performansi.

Song et al. [80] su koristili GA koji je zasnovan na nauci ontologije, gde algoritam može da radi na samoorganizujući način i da reši problem grupisanja tekstualnih dokumenata. Autori su koristili koncepte ontologije, koncepte zasnovane na tezaurusu i koncepte zasnovane na korpusu. U studiji su predstavljene dve hibridne šeme koje koriste različite metrike sličnosti, uključujući metriku zasnovanu na tezaurusu i transformisanu metriku zasnovanu na indeksu latentne semantike (eng. *latent semantic indexing*) koje je nadmašila tradicionalnu metriku sličnosti. U eksperimentima je korišćen *Reuters* skup podataka i rezultati eksperimenata pokazuju da predloženi metod za klasterovanje tekstualnih dokumenata efikasniji od K-srednjih vrednosti i originalnog GA.

Lee et al. [81] su napravili višeciljni GA za pristup klasterovanja tekstualnih dokumenata kako bi smanjio nagib lokalne najmanje vrednosti. U ovoj studiji je korišćen *Reuters* tekstualni skup podataka i rezultati pokazuju da je višeciljni GA nadmašio tri uporedne tehnike za 20%. Uporedne metode u studiji su bili algoritam K-srednjih vrednosti, GA(DBC) i MOGA(DBC-DBE).

Chun-hong et al. [82] su uveli efikasnu metodu za klasterovanje tekstualnih

dokumenata koja je zasnovana na GA i latentnoj semantičkoj analizi (eng. *latent semantic analysis - LSA*). Predloženi metod se bavi sa problemom vektorskog modela prostora i algoritmom K-srednjih vrednosti. Metod je testiran i upoređen sa VSM korišćenjem korpusa sažetka i podataka sa platforme jezičke tehnologije. Dobijeni rezultati studije potvrđuju efikasnost metode u pogledu metrike odziv i preciznosti.

Shi i Lee [83] su predložili metodu CGAM (*Clustering by Genetic Algorithm Model*) za grupisanje tekstova koristeći genetski algoritam i algoritam K-srednjih vrednosti. U predloženom pristupu, GA se koristi kao ciljna funkcija, a K-srednjih vrednosti kao princip konvergencije. Ovaj metod je testiran na kineskom i engleskom tekstualnim skupovima podataka. Predloženi pristup grupisanja kineskog teksta konstruiše naprednu tehniku selekcije početnih GA centara. Rezultati simulacije su pokazali superiornost nad metodama standardnog GA i K-srednjih vrednosti.

Casillas et al. [84] su predstavili GA za grupisanje tekstualnih dokumenata sa pristupom uzorkovanja radi smanjenja vremenske složenosti. GA izračunava aproksimaciju najbolje vrednosti broja klastera  $k$  i rešava problem optimalnog grupisanja tekstualnih dokumenata u klasterima. Performanse ovog pristupa su procenjeni korošćenjem skupova podataka iz rezultata internet pretraživača. Predloženi metod pokazuje da GA sa uzorkovanjem postiže grupisanje u optimalnom vremenu.

Divyashree i Rayar [85] su koristili GA i K-srednjih vrednosti sa kosinusne sličnosti za grupisanje tekstualnih dokumenata. Predstavljeni rad je evauliran pomoću metrike F-mere, čistoće i entropije. Predstavljen metod generiše bolje klastere u komparaciji sa tradicionalnim algoritmom K-srednjih vrednosti.

Karaa et al. [86] su predložili novu metodu za grupisanje sažetaka MEDLINE baze podataka, koja je najveća svetska baza podataka o biomedicinskoj literaturi. Autori su kombinovali GA sa aglomerativnim algoritmom i VSM. Eksperimentalni rezultati pokazuju da je predloženi pristup primenljiv na bilo koji tekstualni skup podataka i bilo koju vrstu elektronskog dokumenta.

Garg i Gupta [87] su koristili pristup grupisanja tekstualnih dokumenata u kome GA se koristi za određivanje početnih vrednosti centroida klastera. Tehnika K-srednjih vrednosti se zatim koristi za kreiranje klastera. Rezultati grupisanja na standardnim skupovima podataka za klasterovanje tekstova, kao što su *20 Newsgroups* i *Classic*, pokazuju da je predložena tehnika nadmašila tehniku K-srednjih vrednosti u smislu

tačnosti.

Mustafi et al. [88] su predstavili novu tehniku za grupisanje tekstualnih dokumenata zasnovanu na GA i koristeći heuristiku najbližeg suseda. Da bi se poboljšao razdvajanje klastera, predstavljena tehnika koristi ponderisanu kombinaciju različitih kriterijuma ciljne funkcije uz uticaj koncepta najbližeg suseda. Predloženo je da ciljna funkcija daje veću težinu tradicionalnim metrikama na udaljenosti unutar i između klastera. Efikasnost predloženog metoda je testirana korišćenjem skupova podataka *BBC* i *20 Newsgroups* i upoređena je sa algoritmom K-srednjih vrednosti.

### 4.3 Klasterovanje tekstualnih dokumenata algoritmom diferencijalne evolucije

Al-Jadir et al. [89] su razvili pristup ACMDHS (*Adaptive Crossover Memetic Differential Harmony Search*) za grupisanje tekstualnih dokumenata. ACMDHS je zasnovan na ukrštanju algoritma diferencijalne evolucije (eng. *Differential evolution - DE*) [90] sa algoritmom traženje harmonije (eng. *Harmony search - HS*) za bolju eksploataciju. Predloženi metod postiže visoko konkurentne rezultate u poređenju sa drugim metodama.

Mustafa et al. [91] su predstavili algoritam memetičke diferencijalne evolucije za rešavanje problema grupisanja teksta, kombinacijom strategija DE mutacije sa memetičkim algoritmom (eng. *Memetic algortihm - MA*) [92]. Ova hibridizacija ima za cilj da poboljša mogućnosti eksploatacije i eksploracije algoritma, kao i kvalitet grupisanja teksta. Na osnovu metrike AUC, F-mere i satističke analize, rezultati simulacije korišćenjem standardnih skupova podataka *CSTR*, *Tr41*, *Tr23*, *Tr12*, *Tr11*, i *Oh15* pokazali su da predložena tehnika nadmašio druge algoritme za grupisanje.

Saini et al. [93] simulirali su grupisanje kao problem sa više prikaza i pristupili su sa stanovišta optimizacije. Nekoliko procena zasnovanih na metrikama sintaksične i semantičke sličnosti korišćeno je da se izvrši grupisanje. U poređenju sa postojećim pristupom, predloženi pristup uključuje tri nove procene zasnovane na udaljenosti. Za procenu performanse, izvršene su simulacije na standardnim tekstualnim skupovima podataka *MORESQUE*, *AMBIENT*, i *ODP-2396*, a rezultati su upoređeni sa drugim postojećim tehnikama u smislu F-mere.

#### 4.4 Klasterovanje tekstualnih dokumenata algoritmom optimizacije sivog vuka

Rashaideh et al. su koristili algoritam optimizacije sivog vuka (eng. *Grey wolf optimizer - GWO*) [94] u [95] za rešavanje problema klasterovanja tekstova. Predloženi metod koristi prosečnu udaljenost dokumenata do centra klastera kao ciljnu funkciju za iterativnu optimizaciju udaljenosti između klastera dokumenata. Efikasnost ovog pristupa je demonstrirana korišćenjem standardnog skupa podataka i metrika kao što su tačnost, preciznost i F-mera. Prema dobijenim rezultatima, predložena metoda je nadmašila druge dve *hill-climbing* tehnike koje su predložene u literaturi.

Vidyadhari et al. [96] su predložili optimizator čestica sivog vuka (eng. *Particle grey wolf optimizer - PGWO*) u kombinaciji sa semantičkom obradom teksta za primenu automatskog grupisanja teksta. U ovom pristupu, tekstualni dokumenti ulaze u fazu predprocesiranja i selekciju atributa, nakon toga se pravi biblioteka atributa i pomoću entripije se biraju najvažniji atributi, zatim predloženi PGWO se koristi za generisanje klastera dokumenata. Eksperimenti su sprovedeni na skupovima podataka *20 Newsgroups* i *Reuters*, a rezultati pokazuju da predložena metoda postigla bolju tačnost od ABC, PSO i GWO.

Abasi et al. [97] su predložili binarni optimizator sivog vuka za selekciju atributa za klasterovanje tekstualnih dokumenata (eng. *Binary grey wolf optimizer - BGWO*). BGWO se koristi za selekciju obeležja, a nakon toga se koristi algoritam K-srednjih vrednosti za grupisanje. Efikasnost predloženog pristupa je testiran na skupovima podataka *Tr42*, *Tr12*, *Classic 4*, *CSTR*, *Wap* i *20 Newsgroups*. Rezultati studije pokazuju superiornost predložene metode u odnosu na GA i BPSO u pogledu prosečne čistoće i F-mere.

#### 4.5 Klasterovanje tekstualnih dokumenata algoritmom potrage za kukavicom

Algoritam potrage za kukavicom (eng. *Cuckoo search - CS*) [98] je predložen u [99] za klasterovanje tekstualnih dokumenata koji su zasnovani na vebu i autori su kombinovali algoritam sa algoritmom PSO. Predložen metod je testiran na standardnim tekstualnim skupovima podataka veb dokumenata i rezultati pokazuju dobre

performase u klasterisanju veb dokumenata.

Boushaki et al. [100] su uveli strategiju grupisanja tekstualnih dokumenata koja je zasnovana na poboljšanom CS algoritmu. Strategija lokalne pretrage algoritma CS je poboljšana za rešavanje problema klasterovanja tekstova. Predloženi metod je testiran na skupovima tekstualnih podataka *Tr23*, *Tr12*, *Classic 300*, *Classic 400* i *Iris*. Rezultati metode ukazuju na robusnost u smislu poboljšanja kvaliteta F-mere, čistoće i ciljne funkcije.

Boushaki et al. [101] su predložili CS-LSI metodu, koja kombinuje CS algoritma sa LSI za dinamičku i inkrementalnu tehniku grupisanja tekstova. Autori su koristili LSI šemu da izaberu najrelevantnije atribute iz tekstualnih skupova podataka. CS-LSI je korišćen za automatsko određivanje broja klastera, kao i za otkrivanje izvanrednih dokumenata održavanjem koherentnih klastera, tako što je metrika udaljenosti značajnosti zasnovan na indeksu. Eksperimenti su sprovedeni na skupovima podataka *Review*, *Sports*, *K1b* i *La2* da bi testirali efikasnost predložene tehnike. U komparaciji sa drugim metodama klasterovanja iz literature, dobijeni rezultati ukazuju na superiornost CS-LSI pristupa za klasterisanje.

Boushaki et al. [102] su kreirali hibridni algoritam kombinovanjem algoritma CS i K-srednjih vrednosti za klasterovanje tekstualnih dokumenata. U algoritmu, CS se koristi za inicijalizaciju centroida klastera u algoritam K-srednjih vrednosti da bi rešili problem inicijalizacije centroida u K-means algoritmu. Efikasnost metode je potvrđen korišćenjem referentnog skupa podataka *Reuters* i rezultati su upoređeni sa tri druge hibridne metode. Rezultati eksperimenta pokazuju da novi hibridni algoritam daje dobre rezultate u pogledu čistoće i F-mere.

#### 4.6 Klasterovanje tekstualnih dokumenata algoritmom svica

Mohammed et al. [103] su koristili algoritam svica (eng. *Firefly algorithm* - FA) [36] za klasterovanje tekstualnih dokumenata. Predložili su pristup hijerarhijskog grupisanja teksta koji je uključivao FA i K-srednjih vrednosti. Rezultati su dobijeni na podacima *20 Newsgroups* i pokazuju da predložena metoda rezultira kompaktnijim klasterima od ostalih nemetaheurističkih metoda.

Mohammed et al. [104] su predstavili gravitaciono grupisanje svica (eng. *Gravity firefly clustering* - GF-CLUST) za klasterovanje tekstualnih dokumenata koji koristi

FA za dinamičko grupisanje dokumenata. GF-CLUST može odrediti odgovarajući broj klastera za datu kolekciju teksta, što predstavlja težak problem u grupisanju dokumenata. Predložena tehnika identifikuje dokumente sa velikom snagom kao centre i generiše klastere na osnovu kosinusne sličnosti. Nakon toga se biraju potencijalni klasteri i mali klasteri se spajaju u njih. Za procenu performanse su korišćeni *Reuters* i *TREC* korpusi. Entropija, F-mera i čistoća nadmašuju rezultate postojećih tehnika grupisanja kao što su K-srednjih vrednosti, PSO i praktična opšta metoda stohastičkog klasterisanja (eng. *Practical general stochastic clustering method - pGSCM*).

#### 4.7 Klasterovanje tekstualnih dokumenata algoritmom krda krila

Abualigah et al. [105] su koristili algoritam krda krila (eng. *Krill herd algorithm - KHA*) [106] za grupisanje tekstova sa veba. U studiji je predstavljeno dva pristupa zasnovana na KHA. Prvi pristup koristi originalni KHA i sve njegove operatore, dok drugi zanemaruje genetske operatore prisutne u originalnom KHA. Efikasnost pristupa je procenjena korišćenjem standardnih skupova podataka iz *CSTR*, *Tr23*, *Tr41* i *Reuters*, i dobijeni rezultati su upoređeni sa rezultatima standardnog algoritma K-srednjih vrednosti. Eksperimentalni rezultati su otkrili da je predloženi pristup grupisanja teksta zasnovan na KHA nadmašio algoritam K-srednjih vrednosti u pogledu kvaliteta klastera pomoću mere čistoće i entropije.

Abualigah et al. u studiji [74] su predstavili hibridni algoritam koji kombinuje KHA sa algoritmom traženje harmonije (eng. *Harmony search - HS*) [107]. Cilj hibridizacije je poboljšanje globalne pretrage algoritma KHA ugradnjom operatora globalne pretrage iz HS algoritma. Ova hibridizacija je rezultirao poboljšanjem sposobnosti eksploracije KHA. Razvijena metoda je testirana na različitim standardnim skupovima tekstulanih podataka i eksperimentalni rezultati pokazuju povećanje tačnosti i veću brzinu konvergencije.

Abualigah i Khader [108] su predložili hibridni metod grupisanja teksta zasnovan na poboljšanom KHA. Autori su modifikovali dve varijante standardnog KHA da bi rešili problem klasterovanja tekstualnih dokumenata i u studiji su predstavili tri varijante KHA. Pošto genetski operatori neznatno poboljšavaju sposobnost globalne pretrage u osnovnom KHA, modifikacija se dešava u redosledu genetskih operatora,

gde se procesi ukrštanja i mutacije pozivaju nakon ažuriranja položaja krila pomoću proračuna KHA kretanja. Predloženi hibridni poboljšani algoritam krda krila je dao najbolje rezultate u poređenju sa K-srednjih vrednosti, PSO, HS i GA.

Abualigah et al. [109] su predložili kombinaciju ciljnih funkcija i modifikovanog KHA za rešavanje grupisanje tekstova. Predloženi metod nasleđuje početnu populaciju KHA od algoritma K-srednjih vrednosti, a odluka o grupisanju se zasniva na dve kombinovane funkcije cilja. Tačnost, preciznost, odziv, F-mera predložene metode su procenjeni korišćenjem skupa podataka *LABIAC* i rezultati su upoređeni drugim tehnikama klasterovanja, kao što su HHS, HGS, HPSO, K-srednjih vrednosti ++.

U radu [110] je predstavljen novi pristup za grupisanje tekstualnih dokumenata korišćenjem četiri različite varijante KHA, originalni KHA, modifikovani KHA, hibridni KHA i višeobjektivni KHA. Svaka varijanta je postepeno poboljšanje u odnosu na metodu pre nje. Performanse predloženih pristupa je procenjen korišćenjem standardnih skupova podataka *CSTR*, *SyskillWebert*, *Tr11*, *Tr12*, *Tr32*, *Tr41*, i *Oh15*. Rezultati simulacije pokazuju da višeobjektivna hibridna varijanta nadmašuje druge varijante i nadmašuje druge komparativne pristupe iz literature.

#### 4.8 Klasterovanje tekstualnih dokumenata algoritmom optimizacije društvenih pauka

Chandran et al. u studiji [111] su koristili optimizaciju društvenih pauka (eng. *Social spider optimization - SSO*) [112] za rešavanje problema grupisanje tekstualnih dokumenata. Performanse ove metode je procenjeno korišćenjem skupa podataka *Patent Corpus 500* i dobijeni rezultati su upoređeni sa standardnim algoritmom K-srednjih vrednosti. Statistički rezultati pokazuju da klasterisanje zasnovano na SSO rezultira sa boljim klasterima od K-srednjih vrednosti. U drugoj studiji [113], isti autori su predstavili grupisanje tekstualnih dokumenata zasnovano na SSO koristeći pristup jednog klastera. U ovom pristupu, svaka jedinka se smatra jednim klasterom. U ovom pristupu, svako rešenje se sastoji od dva podmodula: centroid i lista dokumenata koja su bliže centroidu klastera. Efikasnost predložene metode je testiran i validiran korišćenjem standardnih metrika. Rezultati pokazuju da SSO-k-means radi bolje od k-means-SSO i drugih metoda kao što su k-means-PSO, k-means-GA, poboljšan BCO i k-means-ABC.

## 4.9 Klasterovanje tekstualnih dokumenata algoritmom optimizacije kitova

Gopal i Brunda [114] su predložili metod grupisanja teksta koji uključuje algoritam optimizacije kitova (eng. <sup>9</sup> *Whale optimization algorithm - WOA*) [115] i *fuzzy c-means* (FCM) metodu. WOA je prvo korišćen za generisanje centroida klastera, koji zatim prosleđen FCM algoritmu za stvaranje tekstualnih klastera. Koristeći standardne tekstualne skupove podataka *WebKB*, *Re0* i *20 Newsgroups*, izvršena je eksperimentalna analiza kako bi se procenila efikasnot predložene metode grupisanja teksta u smislu čistoće, F-mere, odziva i preciznosti. Dobijeni rezultati pokazuju superiornost predložene metode nad WOA i FCM.

Sawarn i Deepak [116] su predstavili metodu za grupisanje tekstulanih dokumenata baziran na WOA i semantičku sličnost. Prvo, TF-IDF je korišćen za odabir najinformativnijih atributa iz skupa tekstualnih podatka, dok WOA je korišćen za optimizaciju cele strukture kako bi se poboljšala tačnost predložene metode. Predloženi metod je validiran i upoređen sa drugim tehnikama u literaturi korišćenjem skupa podataka o nasleđu *CHIC*.

## 4.10 Klasterovanje tekstualnih dokumenata algoritmom optimizacije kolonije mrava

Cobo i Rocha [117] su koristili <sup>5</sup> *algoritam optimizacije kolonije mrava* (eng. *Ant colony optimization - ACO*) [118] za klasterisanje tekstualnih dokumenata. Predloženo istraživanje koristi tehiku *thesaurus mining* za generisanje vektora atributa nezavisnog od jezika za izračunavanje sličnosti dokumenata napisanih na različitim jezicima. Feromonski tragovi prisutni u ACO se koriste za odabir vrednosti učešća u proceduri grupisanja. Efikasnost ove metode je procenjena korišćenjem skupa podataka istraživačkih radova u različitim domenima na engleskom i španskom jeziku. Rezultati pokazuju da je predstavljena metoda efikasnija od upoređenih tehnika.

Azaryuon i Fakhar [119] su predložili pristup grupisanja tekstova zasnovan na ACO. U predloženoj metodi promena pozicija mrava je modifikovana u cilju poboljšanja originalnog algoritma grupisanja mrava. Efikasnost algoritma je poboljšana fokusiranjem pokrete mrava u algoritmu. Tekstualni dokumenti iz standardnog skupa

podataka *Reuters* su korišćeni za testiranje predloženog metoda. Eksperimentalni rezultati pokazuju da predloženi metod daje bolje prosečne vrednosti u poređenju sa algoritmom K-srednjih vrednosti i originalnim ACO algoritmom.

Nema i Sharma u radu [120] su koristili ACO algoritam zajedno sa tehnikom klasifikacije klastera relevantnosti (eng. *relevance clustering classification technique*) za grupisanje tekstova sa više oznaka. Algoritam kolonije mrava je korišćen za optimizaciju atributa tekstualnih podataka. Predloženi metod je testiran na nekoliko skupova podataka, uključujući *WebKB*, *Yahoo* i *Reuters*. Drugi algoritmi, kao što su MLFRC i rang SVM, korišćeni su za procenu performansi predložene metode. Rezultati eksperimenta pokazuju da je predloženi metod bolji od MLFRC i rang SVM-a.

#### 4.11 Klasterovanje tekstualnih dokumenata algoritmom traženje harmonije

Cobos et al. [121] su predložili metodu koji je zasnovan na algoritmu traženje harmonije (eng. *Harmony search - HS*) [107] koji je hibridizovan sa algoritmom K-srednjih vrednosti. Predloženi hibridni algoritam je testiran na skupovima podataka *Reuters* i *DMOZ*. Dobijeni rezultati ukazuju na to da predloženi metod dao obećavajuće rezultate u poređenju sa drugim sličnim metodama.

Modeliranjem klasterovanja kao problema optimizacije, Forsati et al. [122] su predložili pristup grupisanja zasnovan na algoritmu HS. U istraživanju HS algoritam je kombinovan sa algoritmom K-srednjih vrednosti na tri različita načina. U prvoj metodi, mehanizam za lokalnu pretragu je zamenjen sa mehanizom globalne pretrage. U drugoj metodi, algoritam K-srednjih vrednosti je poboljšan tako što se manje oslanja na inicijalne nasumične selekcije centroida klastera. U trećoj metodi, varijansa populacije je modelovan kao Markov lanac. Učinak predložene metode validiran korišćenjem skupova podataka *Politics*, *TREC*, *DMOZ* kolekcija, *WebAce* i *20 Newsgroups*. Dobijeni rezultati istraživanja su upoređeni sa standardnim K-srednjih vrednosti i GA. Rezultat simulacije pokazuje da predloženi metod zasnovan na HS pronalazi bolje klustere u smislu F-mere, entropije i čistoće.

Devi i Shanmugam [123] su koristili algoritam K-srednjih vrednosti i HS algoritam za efikasno grupisanje tekstova. Predloženi hibridni metod je koristio TF-IDF za

računanje težine atributa, za validaciju tehnike, *Reuters* skup podataka je korišćen.

Devi i Shanmugam [124] su predstavili još jedan metod grupisanja tekstualnih dokumenata koji je baziran na HS algoritmu u konjukciji sa metodom faktorizacije teorije. Prema rezultatima eksperimenta, teorija faktorizacije može poboljšati originalni HS algoritam.

#### 4.12 Klasterovanje tekstualnih dokumenata optimizatorom multiverzuma

Optimizator multiverzuma (eng. *Multiverse optimizer - MVO*) [125] je korišćen u [126] za klasterovanje tekstualnih dokumenata. U ovom radu autori su prilagodili originalni MVO uključivanjem operatora susedstva (eng. *neighborhood operator*) u cilju mogućnosti poboljšanja globalnog istraživanja algoritma. Prilagođeni algoritam je nazvan kao optimizator multiverzuma zasnovan na linku (eng. *Link-based multiverse optimizer - LBMVO*). *CSTR*, *20 Newsgroups*, *Tr12*, *Tr41*, *Wap*, *Classic 4* skupovi podataka su korišćeni za evaulaciju LBMVO-a. Algoritam je upoređen sa drugim metaheurističkim i nemetaheurističkim metodama (*agglomerative*, DBSCAN, K-srednjih vrednosti++, K-srednjih vrednosti, PSO, GA, HS, KHA, CMAES, COA, i standardni MVO). Rezultati su pokazali superiornost LBMVO-a u odnosu na ostale metode.

Abasi et al. [127] su predložili hibridnu šemu za klasterovanje dokumenata koja kombinuje MVO algoritam sa algoritmom K-srednjih vrednosti. U ovom radu, algoritam K-srednjih vrednosti se koristi za generisanje inicijalne populacije, nakon toga se koristi MVO algoritam za poboljšanje kvaliteta rešenja kao i pronalaženje najboljeg optimalnog rešenja. Predloženi metod je testiran na standardnim tekstualnim skupovima podataka, *CSTR*, *20 Newsgroups*, *NIPS 2015*, *AAAI*, *Tr411*, *Tr12*, i *Wap* podacima. Za evaulaciju kvaliteta su korišćene mere stope greške, tačnost, entropija, odziv, preciznost i F-mera. Eksperimentalni rezultati su potvrdili efikasnost predložene metode u poređenju sa drugim metaheurističkim algoritmima, kao što su KHA, PSO, HS, GA, H-GA, H-PSO, originalni MVO i metodama K-srednjih vrednosti++, K-srednjih vrednosti, DBSCAN i spektranlom metodom.

Abasi et al. u radu [128] su koristili MVO za problem klasterovanja tekstualnih dokumenata. U ovoj studiji, klasterovanje tekstualnih dokumenata se posmatra kao

diskretni problem i funkcija cilja je mera sličnosti koja je zasnovana na Euklidskoj udaljenosti. Ova studija je evaulirana i upoređena sa drugim savremenim tehnikama i dobijeni rezultati ukazuju na efektivnost ove metode u odnosu na druge algoritme.

#### 4.13 Drugi metaheuristički algoritmi za klasterovanje tekstualnih dokumenata

Mirhossein [129] je predložio GSA-KHM pristup grupisanja tekstualnih dokumenata baziran na algoritmu gravitacionog pretraživanja (eng. *Gravitational search algorithm* - *GSA*) [130] i K-harmoničkim sredinama (eng. *K-harmonic means* - *KHM*). Predloženi GSA-KHM algoritam prvo primenjuje KHM algoritam na skupu podataka i dobijeni centroidi klastera se koriste kao rešenja u inicijalnoj populaciji GSA. Predloženi metod je testiran korišćenjem tekstualnih skupova podataka iz *Reuters*, i drugih standardnih skupova podataka iz UCI repozitorijuma mašinskog učenja, kao što su *Iris*, *Wine*, *Glass*, *CMC* i *Cancer*. Eksperimentalna analiza otkriva da GSA-KHM radi bolje od drugih metoda u primeni grupisanja podataka.

Vidyadhari et al. [96] su predstavili pristup za automatsko grupisanje teksta baziran na semantičkoj obradi teksta i poboljšanjem algoritma za optimizaciju roja mačaka (eng. *Cat swarm optimization* - *CSO*) [131]. U ovom pristupu, ulazni dokumenti se prvo prelaze kroz fazu predprocesiranja, koja obezbeđuje odgovarajuće ključne reči za izdvajanje obeležja i grupisanje. Nakon toga, odgovarajuće ključne reči se primenjuju na *wordnet* ontologije za određivanje hiponima i sinonima svake ključne reči. Frekvencija se zatim izračunava za svaku ključnu reč koja se koristi za modeliranje biblioteke tekstualnih karakteristika. Za određivanje najvažnijih atributa se koristi entropija. Performansa predložene tehnike je testirana korišćenjem skupova podataka *20 Newsgroups* i *Reuters*, i rezultati su upoređeni sa ABC, PSO i GA.

Thirumoorthy i Muneeswaran [132] su predložili hibridni algoritam optimizacije JAYA (eng. *JAYA optimization* - *JO*) [133] sa pristupom ukrštanja i mutacije za grupisanje tekstualnih dokumenata. Za evaulaciju kvaliteta rešenja je korišćen siluet indeks (eng. *silhouette index*). Predloženi metod je testiran i upoređen sa standardnim algoritmom K-srednjih vrednosti, GA, CS, PSO, FA, i GWO tehnikama na standardnim skupovima podataka za klasterovanje. Eksperimentalni rezultati pokazuju da je predloženi pristup po kvalitetu nadmašio druge pristupe.

Abasi et al. [134] su predložili novi pristup hibridne optimizacije za efikasno klasterovanje tekstualnih dokumenata kombinacijom (eng. *Salp swarm algorithm* - SSA) [135] i algoritam penjanja (eng. *Hill climbing algorithm* - HC). Cilj ove hibridizacije je poboljšanje kvaliteta početnog rešenja i SSA u smislu njegove lokalne pretrage i stope konvergencije kod optimalne podele klastera. Predloženi hibridni pristup je testiran na šest standardnih tekstualnih skupova podataka i dva skupa podataka iz naučnih članaka. Ishod simulacije otkriva da je predloženi hibridni metod poboljšao rešenja u odnosu na druge pristupe koji se koiriste za grupisanje tekstova, kao što su DBSCAN, K-srednjih vrednosti ++, aglomerativni, spektralni metod i K-srednjih vrednosti, kao i drugi metaheuristički metodi SSA, PSO, KHA, HS, GA, COA, CMAES, i MVO u smislu odziva, F-mere, entropije, tačnosti, čistoće i stope konvergencije.

Perumal i Mathivanan[136] su razvili intucionistički fuzzy klastering tipa 2 i algoritam optimizacije galebova (eng. *Seagull optimization algorithm* - SOA) [137] za klasterovanje tekstualnih dokumenata i detekciju tema. Naziv metode je IFCSOA tipa 2. Predložena metoda je validirana korišćenjem skupa podatka *20 Newsgroups* i upoređen sa algoritmom K-srednjih vrednosti FCM, FCM-GA i FCM-PSO.

## 5 Klasterovanje tekstualnih dokumenata poboljšanom hibridnom metodom metaheuristike i algoritmom nenadgledanog mašinskog učenja

Ovaj odeljak daje opis problema klasterovanja tekstualnih dokumenata, koji je predmet ove distertacije, predstavlja originalni algoritam optimizacije voćne mušice (eng. *Fruit-fly optimization* - *FFO*), koja je glavna komponenta predložene hibridne metode, daje opis prednosti i nedostataka originalnog FFO-a, zatim su predstavljeni ostali algoritmi i komponenti koji su ugrađeni u predloženu hibridnu metodu, originalni algoritam svica (eng. *Firefly algorithm* - *FA*), mehanizam učenja zasnovano na opozici (eng. *Opposition-based learning* - *OBL*), kao i algoritam K-srednjih vrednosti. Na kraju odeljka je predstavljena tačna procedura predloženog hibridnog algoritma, kao i adaptacija predložene metode za problem klasterovanja tekstualnih dokumenata. Naziv novog poboljšanog hibridnog algoritma je hibridna poboljšana voćna mušica svitac (eng. *Hybrid enhanced fruit-fly firefly* - *HEFFF*).

### 5.1 Problem klasterovanja tekstualnih dokumenata

U sadašnjoj, modernoj eri, grupisanje tekstualnih dokumenata (eng. *Text document clustering* - *TDC*) predstavlja veoma značajnu oblast istraživanja, koja se rapidno razvija. TDC je suštinski problem nenadgledanog učenja, jer govori o podeli podataka u nepoznate oblasti. TDC kao polje rudarenje tekstualnih podataka omogućava organizovanje velikih količina tekstualnih podataka. TDC metode se koriste za klasifikaciju dokumenata u srodne grupe određenih kategorija ili specifičnih tema, tako što dokumenti unutar klastera imaju visoku sličnost, a sa druge strane nisku sličnost sa drugim klasterima. Drugim rečima, slični dokumenti se alociraju za sličan klaster, dok različiti dokumenti dodeljuju za različite klastere. Sličnosti i raznolikosti se procenjuju na osnovu vrednosti atributa koji opisuju dokumente.

TDC predstavlja NP-težak problem za pronalaženje klastera u heterogenim dokumentima kroz minimizaciju cilja funkcija.

Pre primene tehnike klasterovanja potrebno je izvršiti prethodnu obradu podataka. Koraci predprocesiranja tekstualnih podataka su:

- tokenizacija,
- uklanjanje zaustavnih reči,
- izvođenje korena i lematizacija
- konverzija tekstualnih podataka u numerički oblik.

**Tokenizacija** U procesu tokenizacije dokumenata zasnovanih na tekstu, svaki dokument u korpusu (zbiru svih dokumenata) se konvertuje u listu tokena (jedinica). U slučaju rečenice, token bi bio reč, a u slučaju reči, tokeni su slova. Na primer nakon procesa tokenizacije rečenice "*Ovo je primer tokenizacije.*", list tokena je: ["*ovo*", "*je*", "*primer*", "*tokenizacije*"].

**Uklanjanje zaustavnih reči** U sledećem koraku predprocesiranja tekstualnih podataka, opšte, uobičajene reči, kao što su članovi, veznici, itd. se uklanjaju sa liste tokena. Primenom procesa uklanjanja zaustavnih reči na gornjem primeru, rezultat bi bio sledeći: ["*primer*", "*tokenizacije*"], gde zaustavni reči "*ovo*" i "*je*" su uklonjeni sa liste tokena. Uklanjanje zaustavnih reči smanjuje veličinu vokabulara što rezultira bržom komputacijom.

**Korenovanje i lematizacija** U procesu korenovanja (eng. *stemming*) i lematizacije (eng. *lemmatization*), reči su normalizovane, što znači da se reči svode na svoj korenski oblik. Razlika između korenovanja i lematizacije je u tome što se u izvođenju korena uklanjaju poslednjih nekoliko slova, dok se kod lematizacije uzima u obzir kontekst i reč se pretvara u njen osnovni, izvorni oblik (lema). Na primer reči "*sam*", "*si*", "*je*", "*smo*", "*ste*", "*su*", nakon primene lematizacije dobijaju njihov izvorni oblik "*biti*".

**Konverzija tekstualnih podataka u numerički oblik** Poslednji korak u predprocesiranju tekstualnih podataka je pretvaranje reči u numerički oblik za izgradnju vokabulara koji sadrži sve tokene, ovaj proces se naziva vektorizacija. Postoje različite tehnike vektorizacije, a terminska frekvencija inverzna frekvencija dokumenata (eng. *Term Frequency-Inverse Document Frequency - TF-IDF*) je jedan od najčešće korišćenih pristupa. Jednačina (13) predstavlja vektor težine ( $w$ )  $i$ -tog dokumenta ( $d$ ):

$$d_i = \{w_{i,1}, w_{i,2}, w_{i,3}, \dots, w_{i,j}, \dots, w_{i,t}\} \quad (13)$$

Težina termina dokumenata se računa prema formuli (14), gde termin je označeno sa  $i$ , određeni dokument sa  $j$ .  $TF(i, j)$  označava učestalost  $i$ -tog termina u dokumentu  $j$ .  $DF(i)$  se odnosi na broj dokumenata koji sadrži termin  $i$ , dok je  $d$  ukupan broj dokumenata.

$$w_{i,j} = TF - IDF(i, j) = TF(i, j) \times \left( \log \frac{d}{DF(i)} \right) \quad (14)$$

Model vektorskog prostora (eng. *Vector space model* - *VSM*) koji se sastoji od  $n$  dokumenata i  $t$  termina je predstavljen sledećim izrazom (15).

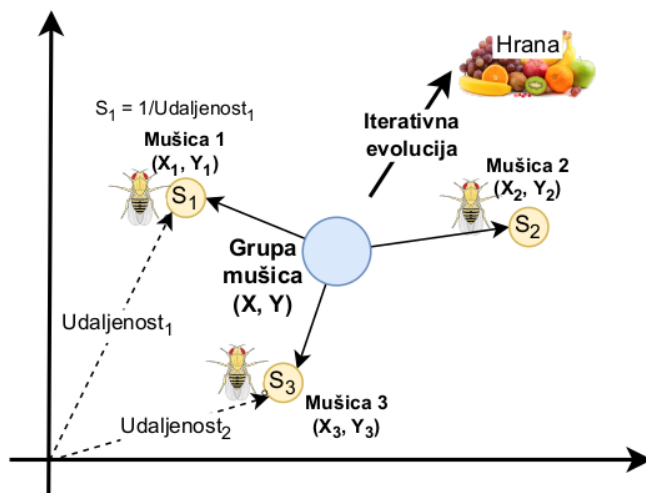
$$VSM = \begin{bmatrix} w_{1,1} & w_{1,2} & \dots & w_{1,t-1} & w_{1,t} \\ \dots & \dots & \dots & \dots & \dots \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ w_{n-1,1} & w_{n-1,2} & \dots & w_{n-1,t-1} & w_{n-1,t} \\ w_{n,1} & w_{n,2} & \dots & w_{n,t-1} & w_{n,t} \end{bmatrix} \quad (15)$$

## 5.2 Algoritam optimizacije voćne mušice

Neki od uobičajenih nedostataka metaheurističkih algoritama su to što imaju relativno komplikovane računске procese, mogu biti teški za razumevanje za početnike, i neki od njih imaju previše parametara. Stoga, autor Pan je predložio novi optimizacioni algoritam, algoritam optimizacije voćne mušice (eng. *Fruit-fly optimization* - *FFO*) [34, 35], koji ima za cilj da pojednostavi računski proces, ima jednostavnu strukturu, lak za razumevanje, kao i implementacija koda.

U FFO radu, autor je postavio model finansijskih nevolja (eng. *financial distress model*) pomoću metode opšte regresije neuronskih mreža optimizovana algoritmom voćne mušice (eng. *Fruit fly algorithm optimized general regression neural network* - *FOAGRNN*) za opštu regresiju neuronskih mreža (eng. *General regression neural network* - *GRNN*), višestruku regresiju (eng. *Multiple regression* - *MR*). Mogućnosti predviđanja klasifikacije ove tri metode su upoređene radi provere izvodljivosti primene algoritama FFO na problemima iz realnog sveta.

FFO je metoda za globalnu optimizaciju, koja je inspirisana na ideji ponašanja voćne mušice u potrazi za hranom. Voćna mušica je superiorna u odnosu na druge vrste kada su u pitanju čula i percepcija, posebno u osfrezi i vidu, kao što ilustruje slika 4.



Slika 4: Ilustracija procesa iterativnog traženja hrane u algoritmu voćne mušice

Organi čula mirisa voćnih mušica mogu naći sve vrste mirisa u vazduhu i mogu ih osetiti u udaljenosti od 40 km-a. Nakon što se voćne mušice približe hrani, pomoću dobre vizualne percepcije pronalaze hranu i leti ka tom pravcu.

Originalni algoritam za optimizaciju voćne mušice sastoji se od četiri faze:

- inicijalizacija,
- osfreza,
- evaluacija populacije i
- faza vida u traženju hrane.

**Inicijalizacija** U početku se rešenja (voćne mušice) generišu nasumično unutar datih donjih i gornjih granica, prema jednačini (16), gde  $x_{i,j}$  označava  $i$ -to rešenje, a  $j$  indeks ukazuje na poziciju elementa u  $i$ -tom rešenju,  $lb$  i  $ub$  se odnose na donju i gornju granicu, respektivno, dok  $rand$  je slučajni broj iz uniformne distribucije.

$$x_{i,j} = rand(ub_j - lb_j) + lb_j \quad (16)$$

**Osfreza** Nakon što se populacija inicijalizuje, pozicija svakog rešenja se ažurira u skladu sa fazom prikupljanja hrane osfrezom, gde se rešenje pomera za slučajni korak od trenutne lokacije i formuliše se matematički kao u jednačini (17), gde je nova pozicija označena sa  $x_{i,j}^{(t+1)}$ , trenutno rešenje je  $x_{i,j}^{(t)}$ ,  $rand() \in [-1, 1]$ . Brojač iteracija je označen sa  $t$ .

$$x_{i,j}^{(t+1)} = x_{i,j}^{(t)} \pm rand() \quad (17)$$

Obzirom da je lokacija hrane nepoznata, udaljenost ( $d$ ) od ishodišta (koordinatnog početka) se procenjuje, zatim se izračunava vrednost procene koncentracije mirisa ( $S$ ) čija vrednost je recipročna vrednost udaljenosti.

$$d_{i,j} = \sqrt{x_{i,j}^2} \quad (18)$$

$$S_{i,j} = \frac{1}{d_{i,j}} \quad (19)$$

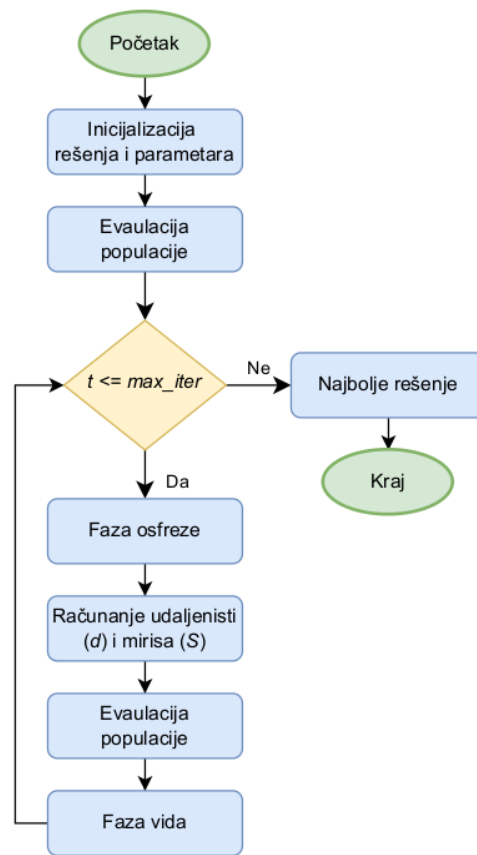
**Evaulacija** Nakon ažuriranja pozicije, izračunava se kondiciona vrednost (vrednost funkcije cilja) za svako rešenje. Supstitucijom vrednosti  $S$  u funkciju cilja ( $f(\cdot)$ ) dobija se pojedinačna koncentracija mirisa i jedinka koja ima najveću vrednost i ujedno predstavlja trenutno najbolje rešenje.

**Vid** Na osnovu vrednosti funkcije cilja odlučuje se da li će nova pozicija ostati ili će se zadržati stara. Ako je kondiciona vrednost novog rešenja bolja od starog, novo rešenje će zameniti staro, u suprotnom, staro rešenje će ostati u populaciji, a novo će biti odbačeno.

Algoritam se prekida kada se dostigne kriterijum zaustavljanja i vraća najbolje rešenje.

Dijagram toka algoritma voćne mušice je predstavljeno na slici 17.

Pseudokod originalnog FFO algoritma je prikazan u algoritmu 3.



Slika 5: Dijagram toka originalnog algoritma voćne mušice

---

**Algoritam 3** Pseudokod originalnog FFO algoritma

---

Inicijalizacija slučajnih položaja populacije prema izrazu (16)  
 Podeševanja brojača iteracija na  $t = 0$  i definisanje kriterijuma za zaustavljanje  
 5 Evaluacija položaja svih jedinki u početnoj populaciji pomoću funkcije cilja  
**while** Kriterijum zaustavljanja nije ispunjen **do**  
   **for**  $i = 1$  to  $N$  **do**  
     Ažuriranje položaja rešenja pomoću postupka pretraživanja izrazom (17)  
     Računanje udaljenosti izrazom (18) i mirisa izrazom (19)  
   **end for**  
   Evaluacija populacije  
   Sortiranje jedinki u populaciji na osnovu koncentracije mirisa  
   Ako je koncentracija mirisa bolja za novo rešenje, zameniti staro rešenje sa novim  
**end while**  
**return** Najbolje rešenje

---

### 5.2.1 Prednosti i nedostaci originalnog algoritma voćne mušice

Glavni prednosti FFO algoritma je njegova relativno jednostavna struktura, algoritam ima mali broj parametara, njegova implementacija je vrlo jednostavna, kao i modifikacija i prilagođavanje algoritma različitim problemima. Međutim, pored ovih značajnih prednosti, algoritam ima i izvesne nedostatke. Rešenja kod FFO algoritma mogu da se „zaglave” u lokalnim minimumima, strategija ažuriranja pozicije je fiksna i zbog toga je algoritam manje efikasniji u procesu eksploatacije. Navedeni nedostaci su uočeni prilikom iscrpnog ekperimentisanja na neograničenim funkcijama.

## 5.3 <sup>12</sup> Algoritam svica

Algoritam svica (eng. *Firefly algorithm - FA*) je razvio Xin-She Yang 2008. godine [36]. Algoritam ima mnogo uspešnih implementacija na različitim problemima optimizacije, kao što je dizajniranje konvolucionih neuronskih mreža [138, 139, 140], procesiranje slika [141], portfolio optimizacija [142], i druge [143, 144, 145, 146, 147].

Pristup FA je motivisan treperevim ponašanjem svica. Algoritam je idealizovan sa tri osnovna pravila:

1. Svaki svitac u populaciji je uniseks, tako da bez obzira na pol, jedan svitac može privlačiti bilo kog drugog.
2. Osvetljenost svica određuje njihov nivo privlačnosti, jer se rastojanje između dva svica smanjuje, svice postaju svetlije i privlačnost se povećava.
3. Osvetljenost definiše kondicionu vrednost (eng. *fitness value*) svica (rešenja).

U početnoj fazi algoritma, rešenja se generišu nasumično unutar donjih i gornjih granica prema sledećoj jednačini:

$$x_{i,j} = lb_j + rand(ub_j - lb_j) \quad (20)$$

gde  $x_{i,j}$  predstavlja  $i$ -to rešenje i odgovarajući  $j$ -ti element, donja i gornja granica su označene sa  $lb_j$  i  $ub_j$ .  $rand$  je slučajni broj iz uniformne raspodele.

Nova lokacija svica određuje se prema izrazu:

$$x_i^{t+1} = x_i^t + \beta_0 r^{-\gamma r_{i,j}^2} (x_j - x_i) + \alpha(rand - 0.5) \quad (21)$$

gde  $x_i^{t+1}$  označava novu poziciju i trenutna pozicija je označeno sa  $x_i^t$ .  $\beta_0$  označava privlačnost na nultom rastojanju,  $x_j$  je svetlije rešenje i trenutno rešenje se kreće ka  $x_j$ .  $\alpha$  je slučajna promenljiva iz Gausove raspodele.

Rastojanje između dva rešenja je izraženo Euklidskom rastojanjem, na sledeći način:

$$r_{i,j} = \|x_i - x_j\| = \sqrt{\sum_{k=1}^d (x_{i,k} - x_{j,k})^2} \quad (22)$$

gde  $r_{i,j}$  se odnosi na rastojanje između rešenja  $i$  i rešenja  $j$ , a dimenzija problema je označena sa  $d$ .

Nakon ažuriranja udaljenosti, vrednost se zamenjuje sledećom jednačinom da bi se izračunala nova vrednost atraktivnosti:

$$\beta(r) = \frac{\beta_0}{1 + \gamma r^2} \quad (23)$$

gde  $\beta_0$  predstavlja atraktivnost kod nultog rastojanja, a  $\beta(r)$  označava novu vrednost atraktivnosti.

Proračun intenziteta svetlosti je definisan na sledeći način:

$$I(r) = \frac{I_0}{1 + \gamma r^2} \quad (24)$$

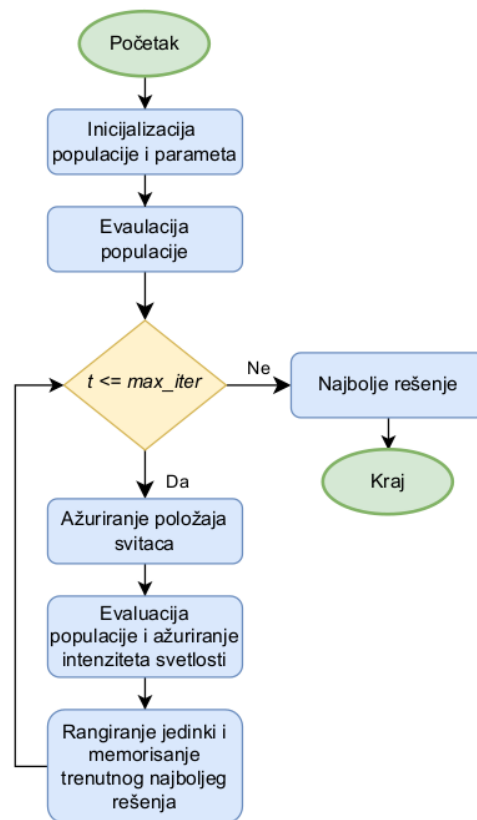
gde je intenzitet svetlosti izvora označen sa  $I_0$ , intenzitet svetlosti na rastojanju  $r$  je označen sa  $I(r)$ , a  $\gamma$  označava koeficijent apsorpcije svetlosti.

Dijagram toka algoritma FA je ilustrovano na slici 6.

Pseudokod FA algoritma je prikazan u algoritmu 4.

## 5.4 Učenje zasnovano na opoziciji

Osnovni koncept učenja zasnovanog na opoziciji (eng. *Opposition-based learning* - *OBL*) je uveo Tizhoosh 2005. godine [148]. Koncept OBL inspirisan suprotnim odnosom između kandidata za rešenje u cilju bržeg pronalaženja boljeg, odnosno optimalnijeg rešenja. Osnovna ideja je da se istovremeno razmatra trenutno rešenje i njena odgovarajuća suprotna procena da bi što efikasnije pronazlazio rešenje koja je bliže globalnom optimumu. OBL mehanizam je u veoma kratkom vremenskom periodu



Slika 6: Dijagram toka algoritma svica

---

**Algoritam 4** Pseudokod originalnog FA algoritma

---

Inicijalizacija slučajnih položaja populacije prema izrazu (20)  
 Podeševanja brojača iteracija na  $t = 0$  i definisanje kriterijuma za zaustavljanje  
 Definisanje koeficijenta apsorpcije svetlosti ( $\gamma$ )  
**while** Kriterijum zaustavljanja nije ispunjen **do**  
   **for**  $i = 1$  to  $N$  **do**  
     **for**  $j = 1$  to  $N$  **do**  
       **if**  $I_j > I_i$  **then**  
         Pomeriti svitac  $i$  prema svicu  $j$  korišćenjem izraza (21)  
       **end if**  
       Evaluacija novog rešenja i ažuriranje intenziteta svetlosti  
     **end for**  
   **end for**  
   Rangiranje svica i sačuvanje najboljeg rešenja  
**end while**  
**return** Najbolje rešenje

---

postao popularan [149, 150] i korišćen u različitim oblastima mekog računarstva (eng. *soft computing*), kao što su metode optimizacije, veštačke neuronske mreže

(eng. *Artificial neural network - ANN*), učenje sa pojačavanjem (eng. *Reinforcement learning - RL*).

Šta je tačno opozicija, odnosno suprotan broj, definisan je u [148, 151] na sledeći način:

### Definicija 1

**Suprotan broj** Neka  $x \in \mathbb{R}$  bude realan broj, definisan u određenom intervalu  $x \in [a, b]$ . Suprotan broj  $\tilde{x}$  definisan kao sledeće:

$$\tilde{x} = a + b - x \quad (25)$$

Analogno tome, može se definisati suprotan broj u višedimenzionalnom slučaju.

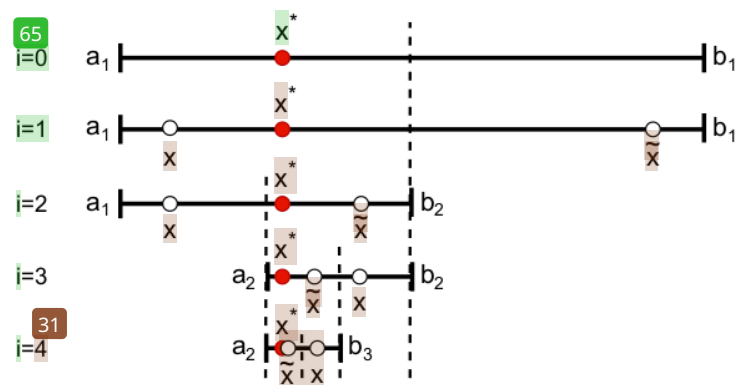
### Definicija 2

**Suprotna tačka u  $n$ -dimenzijonalnom prostoru** Neka  $P(x_1, x_2, x_3, \dots, x_n)$  bude tačka u  $n$ -dimenzijonalnom prostoru, gde  $x_1, \dots, x_n \in \mathbb{R}$  i  $x_i \in [a_i, b_i]$ . Suprotna tačka  $\tilde{P}$  definisan svojim koordinatama  $\tilde{x}_1, \dots, \tilde{x}_n$ , gde:

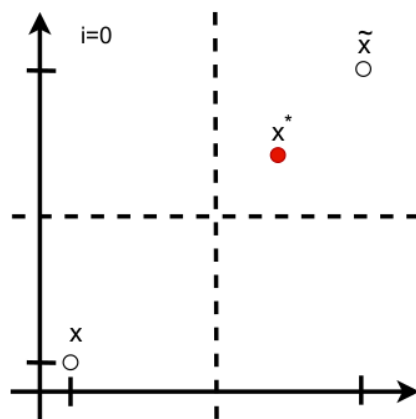
$$\tilde{x}_i = a_i + b_i - x_i \quad \text{gde} \quad i = 1, \dots, n \quad (26)$$

**Učenje zasnovano na opoziciji** Neka je  $f(x)$  funkcija u fokusu, a  $g(\cdot)$  odgovarajuća funkcija evaluacija (na primer fitnes funkcija, funkcija greške, itd.). Ako  $x \in [a, b]$  početna (slučajna) pretpostavka, a  $\tilde{x}$  je njena suprotna vrednost, onda u svakoj iteraciji izračunamo  $f(x)$  i  $f(\tilde{x})$ . Učenje se nastavlja sa  $x$  ako je  $g(f(x)) \geq g(f(\tilde{x}))$ , inače se nastavlja sa  $\tilde{x}$ .

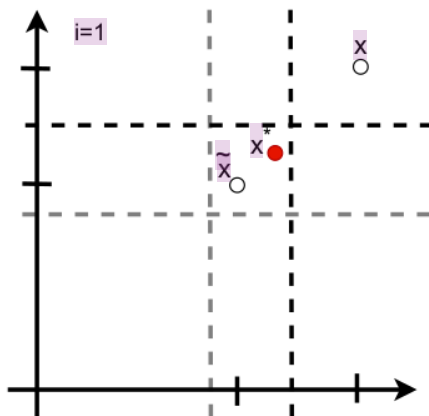
Primer OBL-a u jednodimenzijonalnom prostoru je ilustrovan na slici 7 i u dvodimenzijonalnom prostoru na slikama 8, 9 i 10. Na osnovu toga koja od procena ili suprotna procena je bliža rešenju, interval pretrage se može rekursivno prepoloviti sve dok procena ili suprotna procena ne budu dovoljno blizu postojećem rešenju.



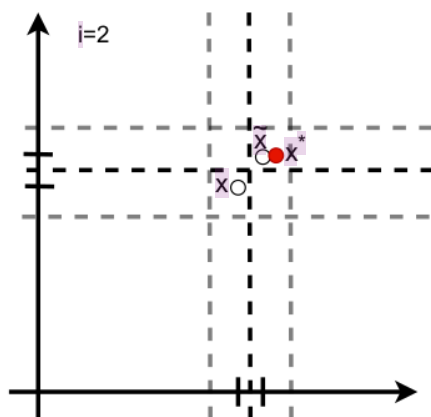
Slika 7: Primer OBL-a u jednodimenzijonalnom prostoru



Slika 8: Primer OBL-a u dvodimenzijonalnom prostoru ( $i = 0$ )



Slika 9: Primer OBL-a u dvodimenzijonalnom prostoru ( $i = 1$ )

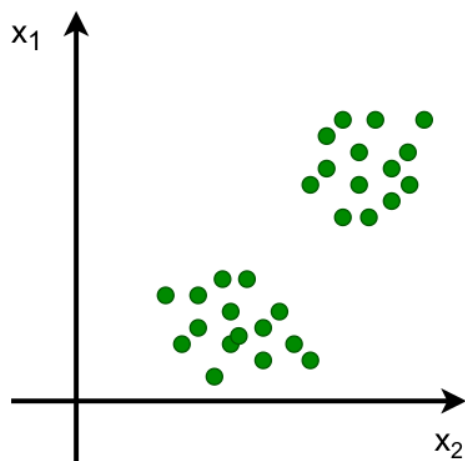


Slika 10: Primer OBL-a u dvodimenzijonalnom prostoru ( $i = 2$ )

## 5.5 K-srednjih vrednosti

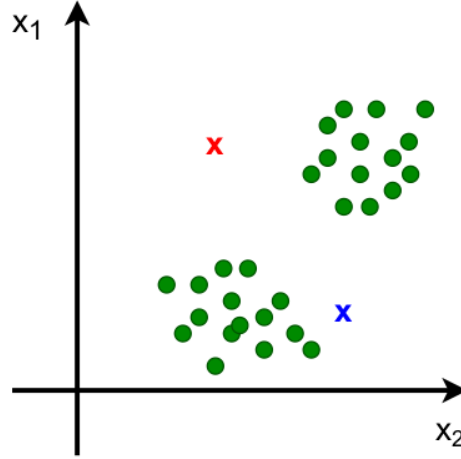
U problemima klasterovanja, imamo podatake koji nisu označene i cilj je da imamo takav algoritam koji će automatski grupisati podatke u koherentne, blisko povezane podskupove podataka, odnosno da otkrije koherentne klastere u skupu podataka. K-srednjih vrednosti (eng. *K-means*) [37] pripada tehnikama nenadgledanog učenja i predstavlja jedan od najpopularnijih i najrasprostranjenijih algoritama za klasterovanje.

Primer procesa algoritma K-srednjih vrednosti je ilustrovana na slikama 11-15. Na ilustrovanom primeru je skup podataka sa dva atributa ( $x_1, x_2$ ).



Slika 11: Primer polaznog skupa podataka

Broj klastera ( $k$ ) se određuje inicijalno, u navedenom primeru, broj klastera je  $k = 2$  jer želimo da grupišemo podatke u dva klastera, centroidi (težište) klastera se inicijaliziraju nasumično.



Slika 12: Primer sa inicijalnim težištima (centroidima) klastera

K-srednjih vrednosti je iterativni algoritam, koji prvo dodeljuje svaki uzorak klasteru koji je najbliži, odnosno razvrstava podatke po klasterima na osnovu udaljenosti od centroida klastera. Udaljenost dva uzoraka se meri korišćenjem Euklidske udaljenosti ili Menhetn rastojanjem. Takođe umesto udaljenosti može da se računaju i sličnosti, na primer kosinusna sličnost ili koeficijent korelacije. Cilj je da se minimizuje funkcija cilja ili funkcija koštanja (eng. *cost function*)  $J$  koja je definisana kao sledeće:

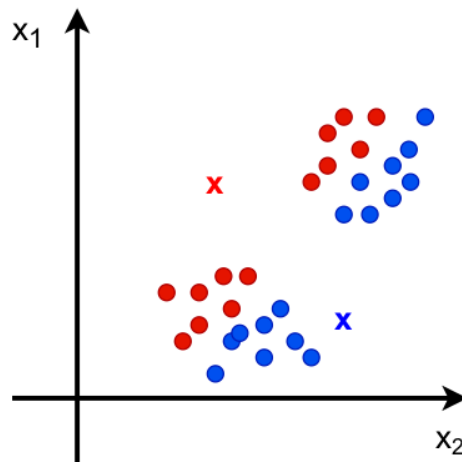
$$J = \sum_{j=1}^k \sum_{i=1}^n \|x_{i,j} - \mu_j\|^2 \quad (27)$$

gde  $J$  označava funkciju cilja,  $k$  je broj klastera,  $x_{i,j}$  je  $i$ -ta instanca u klasteru  $j$ ,  $\mu_j$  je težište klastera  $j$ ,  $\|\cdot\|$  predstavlja Euklidsku normu.

U drugom koraku, centroidi se pomeraju na osnovu prosečne vrednosti uzoraka (instanci) u datom klasteru.

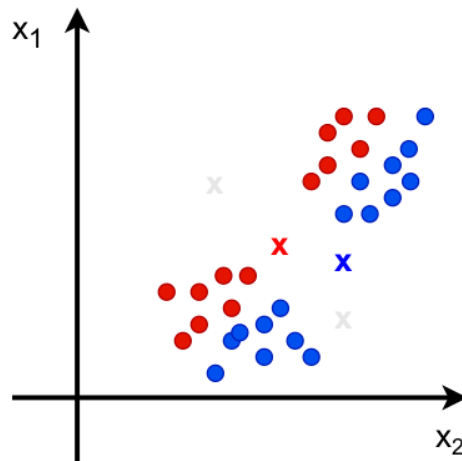
$$\mu_j = \frac{1}{n_j} \sum_{i=1}^{n_j} x_{i,j} \quad (28)$$

gde  $\mu_j$  je težište klastera  $j$ ,  $n$  je broj uzoraka u klasteru  $j$ ,  $x_{i,j}$  je  $i$ -ta instanca u



Slika 13: Primer prvog koraka u iteraciji - dodeljivanje uzoraka klasterima

klasteru  $j$ .

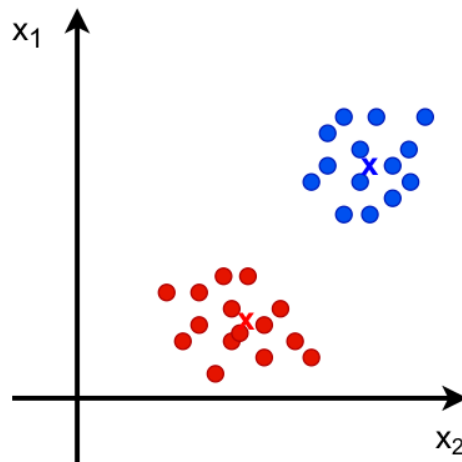


Slika 14: Primer drugog koraka u iteraciji - pomeranje centroida klastera

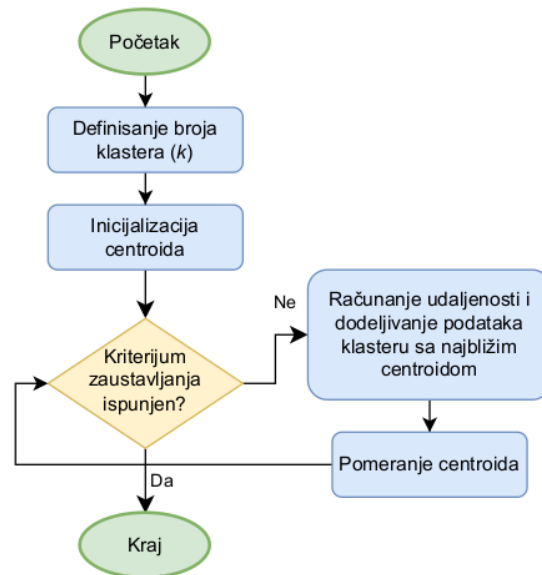
Nakon toga se iterativno ponavljaju oba koraka, razvrstavanje uzoraka po klasterima na osnovu udaljenosti od težišta i pomeranje pozicije centroida na osnovu prosečne vrednosti instanci u klasteru. Koraci se ponavljaju dok algoritam ne dostigne maksimalan broj iteracija ili dok se algoritam ne konvergira, odnosno vrednosti centroida se ne menjaju više.

Dijagram toka algoritma K-srednjih vrednosti je dat na slici 16

Pseudokod algoritma K-srednjih vrednosti je prezentovan u algoritmu 5.



Slika 15: Primer konvergencije algoritma koji zaustavlja proces



Slika 16: Dijagram toka algoritma K-srednjih vrednosti

## 5.6 Hibridna poboljšana voćna mušica i svitac

U cilju eliminisanja nedostataka i poboljšanja faze eksploatacije algoritma FFO, ugrađen je mehanizam pretraživanja iz algoritma svica, koji se odlikuje svojom eksploatacijom. Dodatno, u algoritam je implementiran proces učenje zasnovano na opoziciji (eng. *Opposition-based learning - OBL*) [148] radi boljeg istraživanja prostora za pretragu.

---

**Algoritam 5** Pseudokod algoritma K-srednjih vrednosti

---

**Ulaz:** skup trening podataka od  $m$  broja uzoraka  $X = (x_1, x_2, \dots, x_m)$  i definisanje broja klastera  $k$ .

Nasumična inicijalizacija težište (centroida) klastera  $c$

**repeat**

    Za svaki uzorak izračunati udaljenost od svih težišta klastera i dodeliti najbližem centroidu

    Pomeranje centroida klastera na osnovu prosečne vrednosti uzoraka u datom klasteru.

**until** Ponoviti oba koraka ako nije ispunjen kriterijum zaustavljanja.

Kriterijum zaustavljanja: svaka instanca se već nalazi u najbližem klasteru.

---

Naziv novog hibridnog algoritma je hibridna poboljšana voćna mušica svitac (eng. *Hybrid enhanced fruit-fly firefly - HEFFF*).

HEFFF algoritam, kao i drugi optimizacioni algoritmi, baziran je na populaciji rešenja. Nakon inicijalizacije početne populacije, proces pretraživanja u algoritmu HEFFF se ponavlja kroz iterativni postupak i u svakoj iteraciji se ažuriraju najbolje jedinke dok se ne ispunji kriterijum za zaustavljanje.

Algoritam polazi od slučajnog generisanja jedinki (kandidati rešenja) u populaciji pomoću sledećeg izraza:

$$x_{i,j} = rand(ub_j - lb_j) + lb_j \quad (29)$$

gde  $x_{i,j}$  označava  $i$ -to rešenje, a  $j$  indeks ukazuje na poziciju elementa u  $i$ -tom rešenju i  $rand$  je slučajni broj iz uniformne distribucije. Donje i gornje granice su predstavljene sa  $lb$  i  $ub$ , respektivno.

U parnim iteracijama ( $t$ ) se koristi FFO mehanizam pretrage i učenje zasnovano na opoziciji.

FFO mehanizam pretrage ažurira poziciju rešenja na sledeći način:

$$x_{i,j}^{(t+1)} = x_{i,j}^{(t)} \pm rand() \quad (30)$$

gde je nova pozicija označena sa  $x_{i,j}^{(t+1)}$ , trenutno rešenje je  $x_{i,j}^{(t)}$ ,  $rand() \in [-1, 1]$ . Brojač iteracija je označen sa  $t$ .

Suprotan broj ( $\tilde{x}$ ) kod OBL mehanizma pretrage je definisan sledećom jednačinom:

$$\tilde{x}_{i,j}^{(t+1)} = lb + ub - x_{i,j}^{(t)} \quad (31)$$

gde je nova (suprotna) pozicija označena sa  $\tilde{x}_{i,j}^{(t+1)}$ , a trenutno rešenje je označeno sa  $x_{i,j}^{(t)}$  i  $t$  je brojač iteracija.

Koja od navedena dva mehanizma se koristi u parnim iteracijama, određuje se slučajnim brojem. Ukoliko generisan slučajni broj je manji od 0,5 koristi se FFO mehanizam za ažuriranje pozicije rešenja, inače, ako je broj jednako ili veći od 0,5 u tom slučaju se koristi OBL mehanizam.

U neparnim iteracijama se koristi procedura ažuriranja pozicije rešenja pomoću algoritma svica [36], sledećom jednačinom:

$$x_i^{(t+1)} = x_i^{(t)} + \frac{\beta_0}{1 + \gamma r_{i,j}^2} (x_j - x_i) + \alpha(\kappa - 0.5) \quad (32)$$

gde rastojanje između trenutnog kandidata rešenja ( $x_i$ ) i drugog svica ( $x_j$ ) je označeno sa  $r_{i,j}$  čija vrednost se dobija preko jednačine (33). Privlačnost svica, na multom rastojanju, je označeno sa  $\beta_0$ .  $\gamma$  i  $\alpha$  su kontrolni parametri za randomizaciju.  $\kappa$  se generiše iz Gausove distribucije.

$$r_{i,j} = \|x_i - x_j\| = \sqrt{\sum_{k=1}^D (x_{i,k} - x_{j,k})^2} \quad (33)$$

Za dalje poboljšanje performansi algoritma, ugrađen je dinamički korak za kontrolni parametar  $\alpha$  iz modifikovanog algoritma svica sa prilagodljivim kontrolnim parametrima (eng. *Firefly algorithm with adaptive control parameters - ApFA*) [152]. U strategiji dinamičkog prilagođavanja adaptivnog parametra  $\alpha$ , vrednost se postepeno smanjuje tokom iteracija od početne vrednosti ( $\alpha_0$ ) do minimalne vrednosti ( $\alpha_{min}$ ).

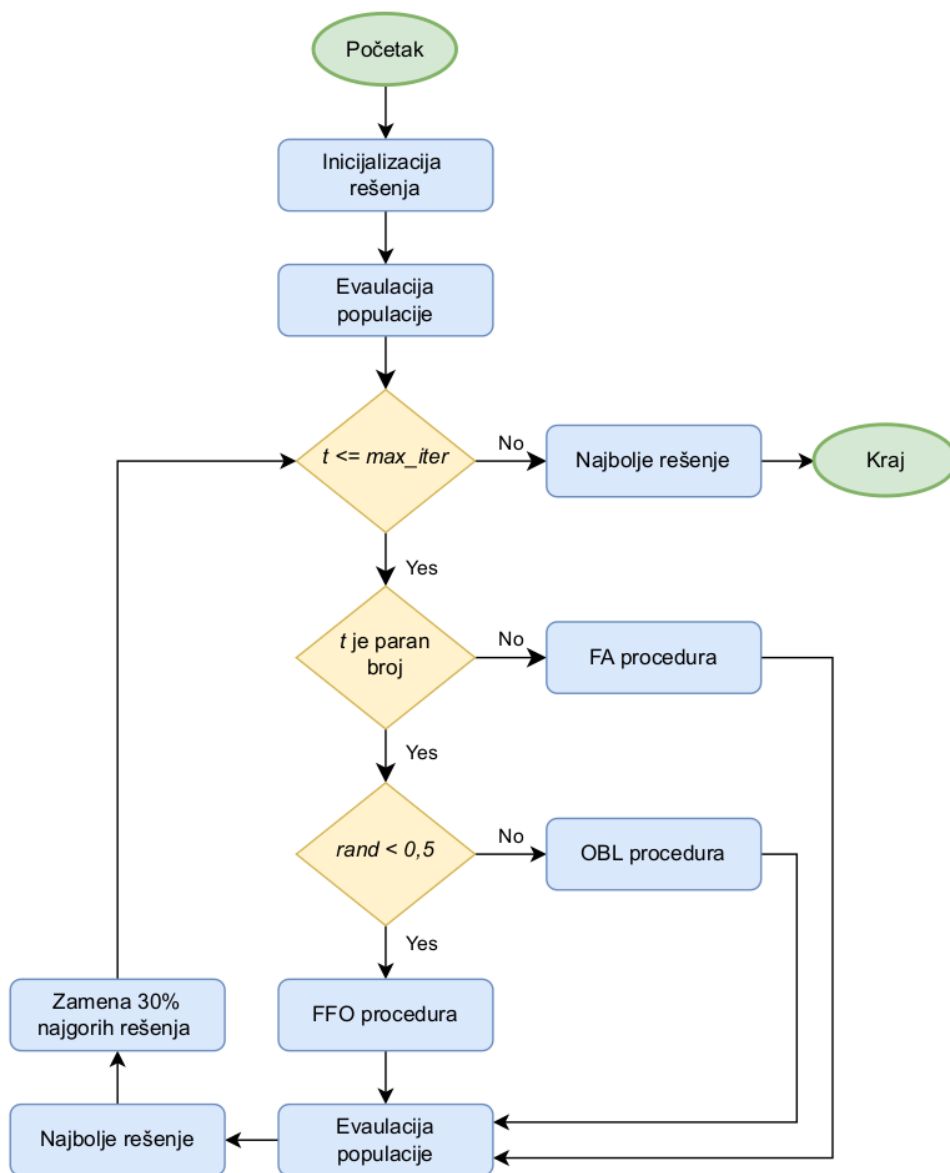
U svakoj iteraciji vrednost za parametar  $\alpha$  se izračunava putem sledećeg izraza:

$$\alpha^{(t+1)} = \alpha^{(t)} \cdot \left(1 - \frac{t}{MaxIter}\right) \quad (34)$$

gde  $t$  predstavlja trenutnu iteraciju,  $MaxIter$  je maksimalan broj iteracija.  $\alpha^{(t)}$  je vrednost  $\alpha$  u iteraciji  $t$ , dok ažurirana vrednost za sledeću iteraciju je označeno sa  $\alpha^{(t+1)}$ .

Dijagram toka HEFFF algoritma dat je na slici 17.

Pseudokod predloženog HEFFF algoritma je prikazan u algoritmu 6.



Slika 17: Dijagram toka predloženog HEFFF algoritma

## 5.7 Klasterovanje tekstualnih dokumenata poboljšanom hibridnom metodom HEFFF i algoritmom K-srednjih vrednosti

Klasterovanje tekstualnih dokumenata ima dve faze, u prvoj fazi se radi predprocesiranje ulaznih tekstualnih podataka, uključujući vektorizaciju, odnosno konverto-

---

**Algoritam 6** Pseudokod predloženog HEFFF algoritma

---

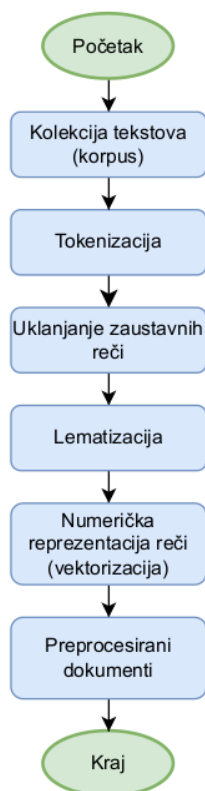
Inicijalizacija slučajnih položaja populacije prema izrazu (16)  
Inicijalizacija FA parametara:  $\beta_0, \gamma, \alpha_0, \alpha_{min}$   
Podeševanja brojača iteracija na  $t = 0$  i definisanje kriterijuma za zaustavljanje  
Evaluacija položaja svih jedinki u početnoj populaciji pomoću funkcije cilja  
**while** Kriterijum zaustavljanja nije ispunjen **do**  
    **for**  $i = 1$  to  $N$  **do**  
        **if**  $t$  je paran **then**  
            **if**  $rand < 0,5$  **then**  
                Ažuriranje položaja rešenja pomoću postupka pretraživanja izrazom (17)  
                iz FFO algoritma  
            **else**  
                Ažuriranje položaja rešenja pomoću postupka pretraživanja izrazom (31)  
                iz OBL procedure  
            **end if**  
        **else**  
            Ažuriranje položaja rešenja pomoću postupka pretraživanja izrazom (32) iz  
            FA algoritma  
        **end if**  
    **end for**  
    Evaluacija položaja svih jedinki. Ako je vrednost funkcije cilja je bolja za novo  
    rešenje, zameniti staro rešenje **5** novim  
    Sortiranje jedinki u populacije **na osnovu vrednosti funkcije cilja**  
    Ažurirati **najb5** je trenutno rešenje  
    Zameniti 30% jedinki u populaciji **sa** najgorim **vrednostima funkcije cilja** generi-  
    sanjem novih slučajnih rešenja  
    **end while**  
**return** Najbolje rešenje

---

vanje reči u numeričke vrednosti, i vrednosti su normalizovane između 0 i 1. U drugoj fazi se radi klasterovanje nad predprocesiranom kolekcijom tekstova, takozvanog korpusa (eng. *corpus*). Klasterovanje se radi pomoću predložene hibridne metode voćne mušice i algoritmom K-srednjih vrednosti. Dijagram toka procesa obrade ulaznih tekstulanih podataka je prikazano na slici 18.

Algoritam polazi od slučajnog generisanja jedinki u populaciji između donje i gornje granice. Obzirom da su podaci normalizovane između 0 i 1, donja granica je 0, a gornja granica je 1.

Jedna jedinka u populaciji predstavlja jedno od mogućih rešenja za klasterovanje podataka korpusa teksta. Dakle, populacija predstavlja rešenje kandidata za klasterovanje tekstualnih podataka. Svako rešenje je kodirano na sledeći način:  $X_i = (C_1, C_2, \dots, C_i, \dots, C_k)$ , gde  $C_i$  predstavlja  $i$ -ti vektor centroida klastera i  $k$



Slika 18: Dijagram toka procesa obrade ulaznih tekstualnih podataka

je broj klastera. Na osnovu mere udaljenosti koja je ujedno i ciljna funkcija, svaki dokument je dodeljen najbližem klasteru.

Ciljna funkcija (rastojanje između dokumenta i centroida) je računato pomoću metrike udaljenosti (Euklidska udaljenost), koja je definisana jednačinom (35):

$$dist(d_j, c_k) = \sqrt{\sum_{i=1}^t |d_{ij} - c_{ik}|^2} \quad (35)$$

gde  $t$  predstavlja broj pojmova, odnosno atributa u dokumentu,  $d$  označava dokument, a  $c$  je centroid.

Nakon inicijalizacije rešenja radi se vrednovanje prema funkciji cilja za svako rešenje i sortiranje vrednosti ciljne funkcije za svaku jedinku, od najbolje vrednosti do najgore, posle toga se dodeljuje svaki dokument klasteru sa najbližim centroidom. Postupak evoluiranja rešenja sa HEFFF algoritmom se iterativno ponavlja kroz unapred definisan broj maksimalne iteracije. Kada se ispuni kriterijum za zaustavljanje

(maksimalan broj iteracija za HEFFF algoritam), primenjuje se algoritam K-srednjih vrednosti na dobijenim pozicijama centra klastera iz HEFFF algoritma.

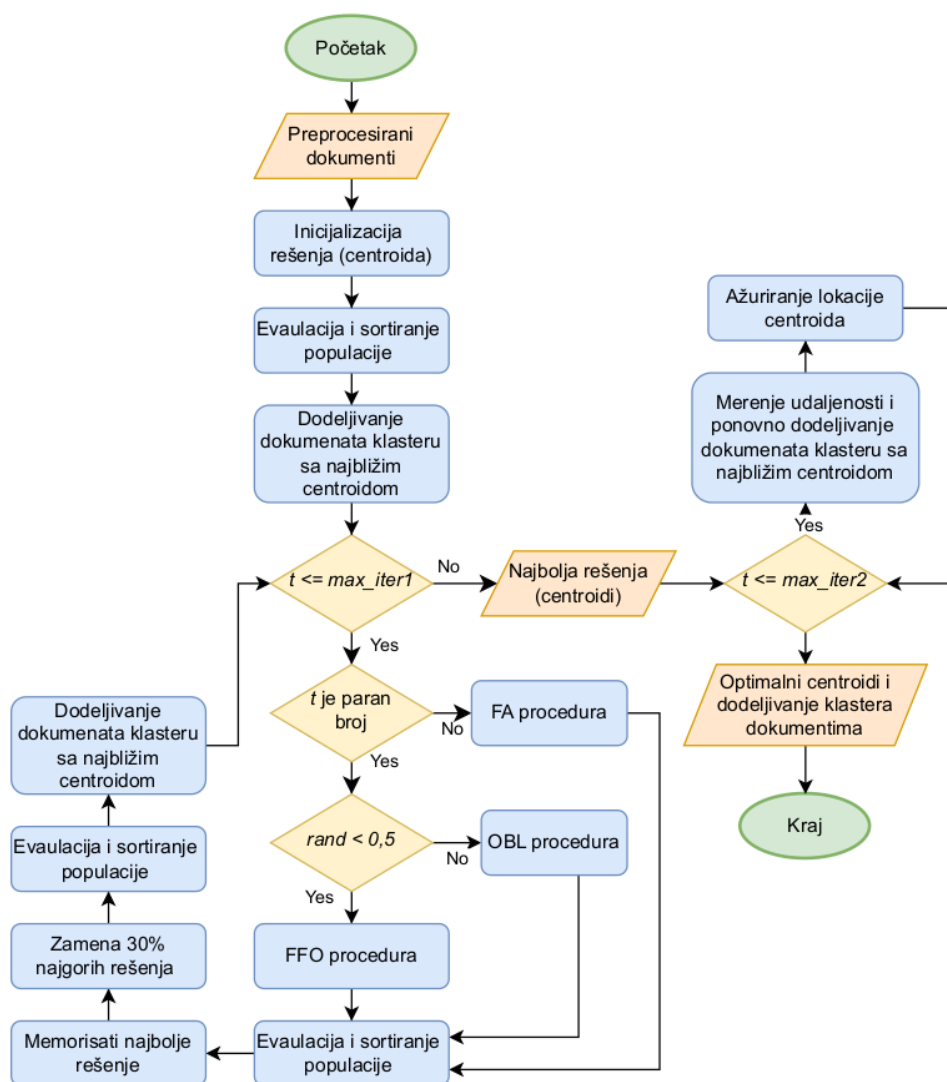
U algoritmu K-srednjih vrednosti se radi ažuriranje lokacije centroida na osnovu sledeće jednačine: (36).

$$f(x) = \frac{\sum_{i=1}^k (\frac{1}{n_i} \sum_j^{n_i} dist(d_{ij}, c_i))}{K} \quad (36)$$

gde ukupan broj klastera označeno sa  $K$ ,  $n_i$  predstavlja broj dokumenata u  $i$ -tom klasteru,  $d_{ij}$  predstavlja  $j$ -ti dokument u klasteru  $i$ . Centroid  $i$ -tog klastera je označen sa  $c_i$ . Funkcija  $dist()$  računa rastojanje između dokumenta  $d_{ij}$  i centroida klastera  $c_i$ .

Dijagram toka predloženog HEFFF algoritma sa algoritmom K-srednjih vrednosti za klasterovanje tekstualnih dokumenata je prikazan na slici 19.

Pseudokod predloženog HEFFF algoritma sa algoritmom K-srednjih vrednosti za klasterovanje tekstualnih dokumenata je prikazan u algoritmu 7.



Slika 19: Dijagram toka predloženog HEFFF algoritma u kombinaciji sa algoritmom K-srednjih vrednosti za klasterovanje tekstualnih dokumenata

---

**Algoritam 7** Pseudokod predloženog HEFFF algoritma sa algoritmom K-srednjih vrednosti za klasterovanje tekstualnih dokumenata

---

**Faza 1.**

Ulaz: korpus

Tokenizacija

Uklanjanje zaustavnih reči

Lematizacija

Vektorizacija

**Faza 2.**

Inicijalizacija slučajnih položaja populacije prema izrazu (16)

Inicijalizacija FA parametara:  $\beta_0, \gamma, \alpha_0, \alpha_{min}$

Podeševanja brojača iteracija na  $t = 0$  i definisanje kriterijuma za zaustavljanje

Evaulacija položaja svih jedinki u početnoj populaciji pomoću funkcije cilja

**HEFFF**

**while** Kriterijum zaustavljanja nije ispunjen **do**

**for**  $i = 1$  to  $N$  **do**

**if**  $t$  je paran **then**

**if**  $rand < 0,5$  **then**

                Ažuriranje položaja rešenja pomoću postupka pretraživanja izrazom (17)  
                iz FFO algoritma

**else**

                Ažuriranje položaja rešenja pomoću postupka pretraživanja izrazom (31)  
                iz OBL procedure

**end if**

**else**

            Ažuriranje položaja rešenja pomoću postupka pretraživanja izrazom (32) iz  
            FA algoritma

**end if**

**end for**

    Evaulacija položaja svih jedinki. Ako je vrednost funkcije cilja je bolja za novo  
    rešenje, zameniti staro rešenje **5** novim

    Sortiranje jedinki u populacije **na osnovu vrednosti funkcije cilja**

    Ažurirati **najb5** je trenutno rešenje

    Zameniti 30% **jedinki u** populaciji **sa** najgorim **vrednostima funkcije cilja** generi-  
    sanjem novih slučajnih rešenja

**end while**

**return** Najbolje rešenje

**5-srednjih vrednosti**

**while** Kriterijum zaustavljanja nije ispunjen **do**

**for**  $i = 1$  to  $N$  **do**

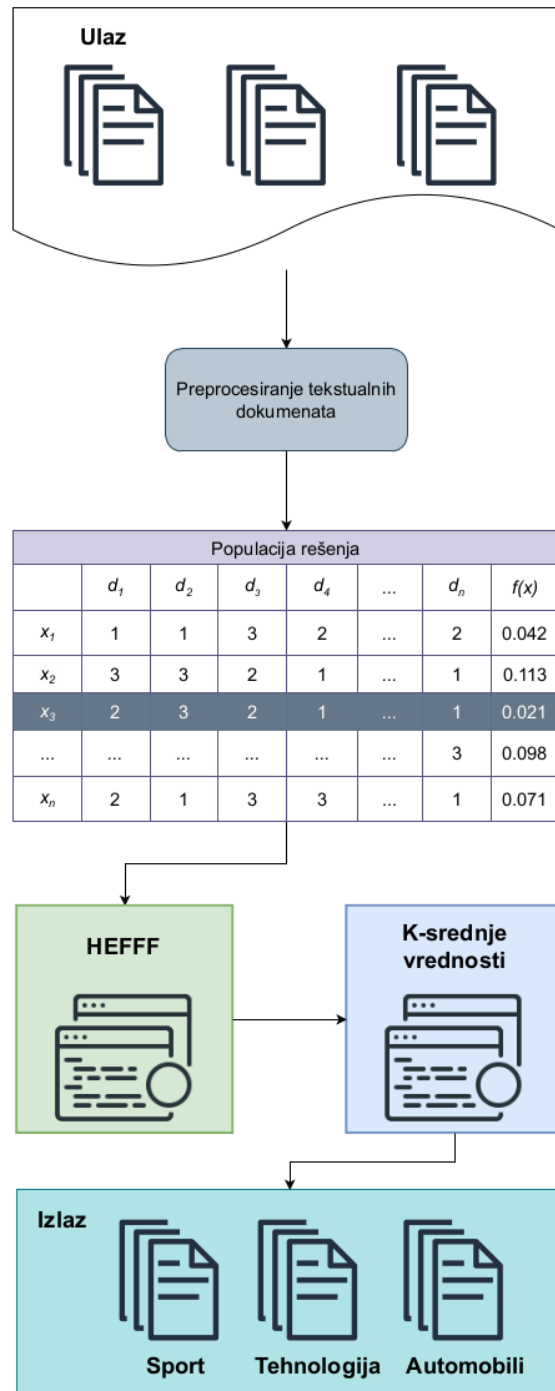
        Ažuriranje položaja rešenja pomoću izraza (36) iz metode K-srednjih vrednosti

**end for**

**end while**

**return** Optimalni centroidi i dodeljivanje klastera dokumentima

---



Slika 20: Dijagram toka klasterovanja tekstualnih dokumenata

## 6 Eksperimentalni rezultati

Predložena HEFFF metoda je prvo validirana na neograničenim funkcijama, kao standardna procedura za validaciju novouvedenih metaheurističkih algoritama, a zatim kombinovana sa algoritmom K-srednje vrednosti i primenjen je za rešavanje problema klasterovanja tekstualnih dokumenata.

### 6.1 Eksperimenti na CEC 2019 benčmark funkcijama

Postoji jaka veza između metaheurističkih algoritama i numeričkih test funkcija, pa je stoga kada se uvodi novi metaheuristički algoritam, potrebno je koristiti benčmark funkcije za validaciju performanse.

Benčmark funkcije su u osnovi optimizacioni problemi koji su formulisani u obliku matematičkih numeričkih funkcija. Najbolje rešenje se krije u velikom broju suboptimalnih rešenja rasprostranjenih po celom prostoru pretrage. Efikasnost metaheuristike se meri sposobnošću globalnog pretraživanja i lokalne konvergencije. Bolja sposobnost globalnog pretraživanja sprečava zaglavljivanje u suboptimalnim lokacijama, a s druge strane efikasna sposobnost konvergencije sprečava propust najboljeg, optimalnog rešenja.

Nakon testiranja na benčmark funkcijama, potrebno je sprovesti uporednu analizu rezultata sa drugim metaheurističkim algoritmima koji su primenjeni na istom skupu problema za dokazivanje efikasnosti algoritma. Dobar skup benčmark funkcija treba da sadrži funkcije sa različitim karakteristikama i treba da bude nepristrasan, uz to, za rigoroznije testiranje takođe funkcije treba da budu rotirane, pomerene i da imaju širi opseg dimenzija za dodatni složenost problemima.

Zbog navedenih, performanse predložene metode su validirane na 10 modernih CEC (*Congress on evolutionary computation*) 2019 funkcijama [153] i dobijeni rezultati su upoređeni sa originalnim FFO i drugih, devet metaheurističkih pristupa (EHOI [154], EHO [155], SCA [156], SSA [135], GOA [157], WOA [158], BBO [159], MFO [160], PSO [67]), gde su simulacije sprovedene pod sličnim uslovima i isti skupovi problema su korišćeni.

### 6.1.1 Definicija CEC 2019 test funkcija

Sve test funkcije su problemi minimizacije definisani na sledeći način:

$$\min f(x), \text{ gde } x = [x_1, x_2, \dots, x_D]^T \quad (37)$$

$D$  je dimanzija problema. Dimenzija prve funkcije je 9, druge funkcije je 16, treće funkcije je 18, a funkcije 4-10 imaju dimenziju 10.

Pomereni globalni optimum  $o_i = [o_1, o_2, \dots, o_D]^T$ , koji je slučajno raspoređen u intervalu  $[-80, 80]^D$ .

Sve test funkcije su skalabilne. Funkcije 4-10 su rotirane i pomerane do  $o$ , dok funkcije 1-3 nisu. Funkcije 4-10 važe za bilo koje  $D > 1$ , a funkcije 1-3 važe za  $D = 2n, n^2, n^3$ , gde  $n = 1, 2, 3$ , itd.

Prostor pretrage za funkciju 1 je  $[-8192, 8192]$ , za funkciju 2 je  $[-16384, 16384]$  i za za funkciju 3 je  $[-4, 4]$ , dok za funkcije 4-10 je isti interval  $[-100, 100]$ .

$M_i$  je matrica rotacije, koja se za svaku podkomponentu generiše posebno i različite matrice rotacije se dodeljuju funkcijama 4-10, dok prve tri funkcije nisu rotirane.

### CEC funkcija 1

#### Storn Čebiševa problem uklapanja polinoma

$$f_1(x) = p_1 + p_2 + p_3 \quad (38)$$

$$p_1 = \begin{cases} (u - d)^2 & \text{ako } u < d, \\ 0 & \text{inače.} \end{cases}$$

$$d = \sum_{j=1}^D x_j (1.2)^{D-j}$$

$$p_1 = \begin{cases} (v - d)^2 & \text{ako } v < d, \\ 0 & \text{inače.} \end{cases}$$

$$v = \sum_{j=1}^D x_j (-1.2)^{D-j}$$

$$p_k = \begin{cases} (w_k - 1)^2 & \text{ako } w_k > 1, \\ (w_k + 1)^2 & \text{ako } w_k < 1, \\ 0 & \text{ina}\check{c}e. \end{cases}$$

$$w_k = \sum_{j=1}^D x_j \left(\frac{k2}{m} - 1\right)^{D-j}$$

$$p_3 = \sum_{k=0}^m p_k$$

$$k=0,1,...,m$$

$$m=32D$$

$$d=72.661 \text{ za } D=9$$

$$F_1(x)=f_1(x)+F_1^* \tag{39}$$

Ova funkcija nije definisana za dvodimenzionalni problem.

**CEC funkcija 2**

**Inverzna Hilbertova matrica**

$$f_2(x) = \sum_{i=1}^{\overline{n}} \sum_{k=1}^{\overline{n}} |w_{i,k}|^{72} \tag{40}$$

$$\begin{matrix} 3 \\ (w_{i,k}) \end{matrix} = \mathbf{W} = \mathbf{HZ} - \mathbf{I}, \text{ gde } \mathbf{I} = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{bmatrix}$$

$$\mathbf{H} = (h_{i,k}), \text{ gde } h_{i,k} = \frac{1}{i+k-1}, \text{ gde } i, k = 1, 2, 3 \dots, n, \text{ gde } n = \sqrt{D}$$

$$\mathbf{Z} = (z_{i,k}), \text{ gde } z_{i,k} = x_{i+n(k-1)}$$

$$F_2(x) = f_2(x) + F_2^*$$

Ova funkcija nije definisana za dvodimenzionalni problem.

#### CEC funkcija 3

Lenard-Jones Minimum Energy Cluster

$$f_3(x) = 12.7120622568 + \sum_{i=1}^{n-1} \sum_{j=i+1}^n \left( \frac{1}{d_{i,j}^2} - \frac{2}{d_{i,j}} \right) \quad (41)$$

$$\begin{matrix} 3 \\ d_{i,j} \end{matrix} = \left( \sum_{k=0}^2 (x_{3i+k-2} - x_{3j+k-2})^2 \right)^3, \text{ gde } n = D/3$$

$$F_3(x) = f_3(x) + F_3^* \quad (42)$$

Ova funkcija nije definisana za dvodimenzionalni problem.

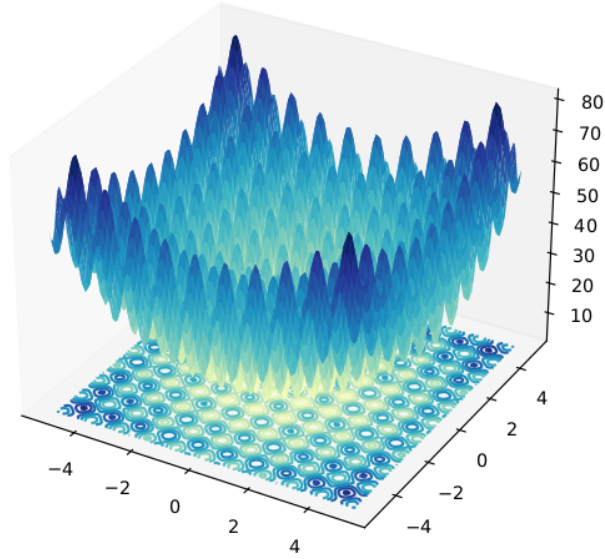
#### CEC funkcija 4

Rastriginova funkcija

$$f_4(x) = \sum_{i=1}^D (x_i^2 - \cos(2\pi x_i) + 10) \quad (43)$$

### Pomerena i rotirana Rastriginova funkcija

$$F_4(x) = f_4(M(x - o_4)) + F_4^* \quad (44)$$



Slika 21: 3D grafikon dvodimenzionalne Rastrigin funkcije

### CEC funkcija 5

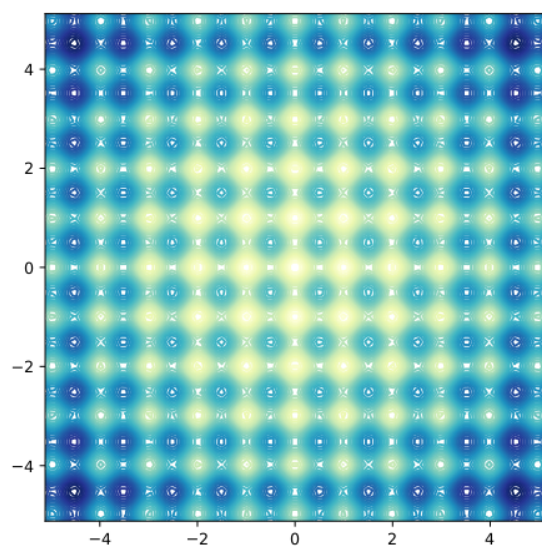
#### Grivankova funkcija

$$f_5(x) = \sum_{i=1}^D \frac{x_i^2}{4000} - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \quad (45)$$

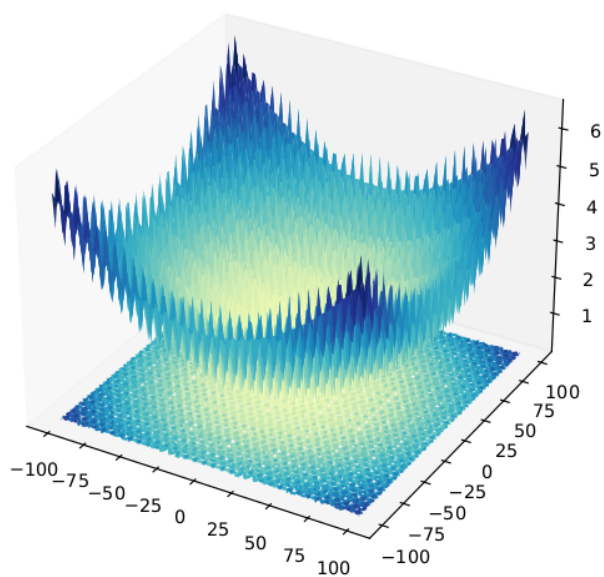
#### Pomerena i rotirana Grivankova funkcija

$$F_5(x) = f_5(M(x - o_5)) + F_5^* \quad (46)$$

### CEC funkcija 6



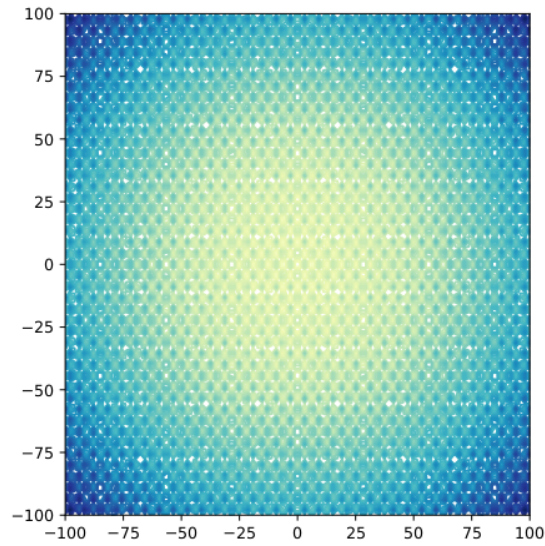
Slika 22: Kontur grafikon dvodimenzionalne Rastrigin funkcije



Slika 23: 3D grafikon dvodimenzionalne Griewank funkcije

### Vajerštrasova funkcija

$$f_6(x) = \sum_{i=1}^D \left( \sum_{k=0}^{k_{max}} [a^k \cos(2\pi b^k (x_i + 0.5))] \right) - D \sum_{k=0}^{k_{max}} a^k \cos(\pi b^k) \quad (47)$$



Slika 24: Kontur grafikon dvodimenzionalne Griewank funkcije

Preporučene vrednosti promenljivih su:  $a = 0.5$ ,  $b = 3$  i  $k_{max} = 20$ .

Pomerena i rotirana Švefelova funkcija

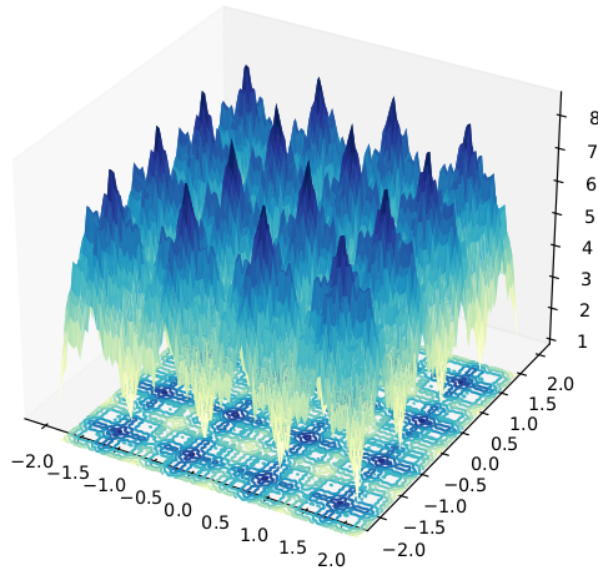
$$F_6(x) = f_6(M(x - o_6)) + F_6^* \quad (48)$$

CEC funkcija 7

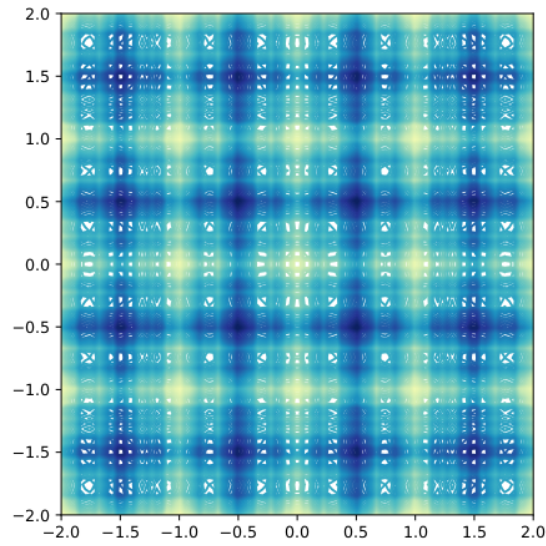
Modifikovana Švefelova funkcija

$$f_7(x) = 418.9829D - \sum_{i=1}^D g(z_i), \text{ gde } z_i = x_i + 420.9687462275036 \quad (49)$$

$$g(z_i) = \begin{cases} z_i \sin(|z_i|^{1/2}) & \text{if } |z_i| \leq 500 \\ (500 - \text{mod}(z_i, 500)) \sin(\sqrt{|500 - \text{mod}(z_i, 500)|}) - \frac{(z_i - 500)^2}{10000D} & \text{if } |z_i| > 500 \\ ((\text{mod}(|z_i|, 500) - 500) \sin(\sqrt{|\text{mod}(z_i, 500) - 500|}) - \frac{(z_i + 500)^2}{10000D}) & \text{if } |z_i| < 500 \end{cases}$$



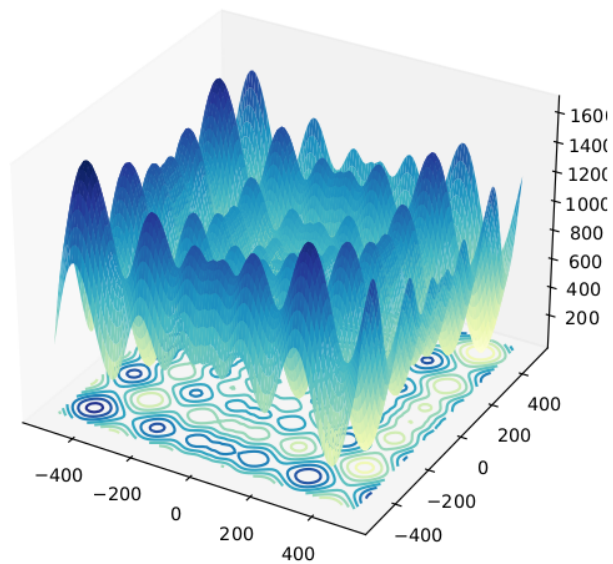
Slika 25: 3D grafikon dvodimenzionalne Weierstrass funkcije



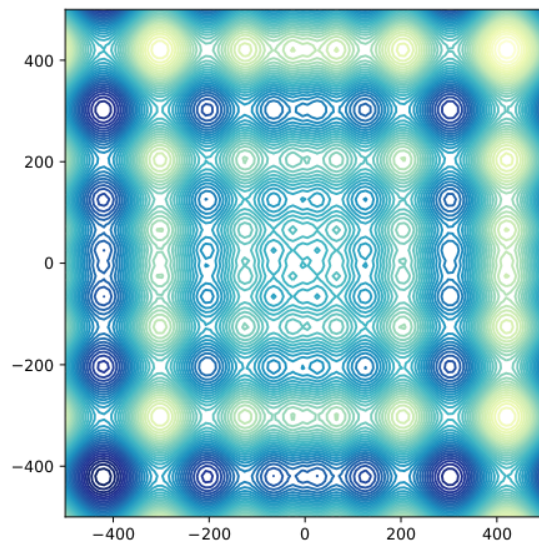
Slika 26: Kontur grafikon dvodimenzionalne Weierstrass funkcije

**Pomerena i rotirana Švefelova funkcija**

$$F_7(x) = f_7(M(x - o_7)) + F_7^* \quad (50)$$



Slika 27: 3D grafikon dvodimenzionalne Schwefel funkcije



Slika 28: Kontur grafikon dvodimenzionalne Schwefel funkcije

**CEC funkcija 8**

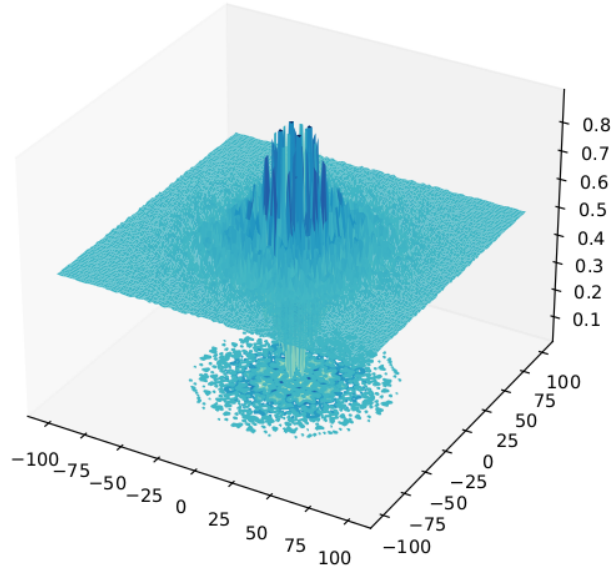
### Proširena Šaferova funkcija F6

$$g(x, y) = 0.5 + \frac{\sin^2(\sqrt{x^2 + y^2}) - 0.5}{(1 + 0.001(x^2 + y^2))^2} \quad (51)$$

$$f_8(x) = g(x_1, x_2) + g(x_2, x_3) + \cdots + g(x_{D-1}, x_D) + g(x_D, x_1)$$

### Pomerena i rotirana Šaferova funkcija F6

$$F_8(x) = f_8(M(0.005(x - o_8))) + F_8^* \quad (52)$$

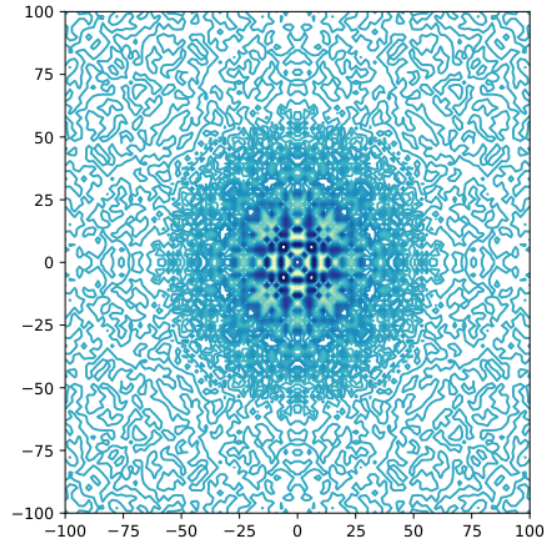


Slika 29: 3D grafikon dvodimenzionalne Schaffer funkcije

### CEC funkcija 9

#### Happy Cat funkcija

$$f_9(x) = \left| \sum_{i=1}^D x_i^2 - D \right|^{1/4} + (0.5 \sum_{i=1}^D x_i^2 + \sum_{i=1}^D x_i) / D + 0.5 \quad (53)$$



Slika 30: Kontur grafikon dvodimenzionalne Schaffer funkcije

Preporučene vrednosti promenljivih su:  $a = 20$ ,  $b = 0,2$  i  $c = 2\pi$ .

#### Pomerena i rotirana Happy Cat funkcija

$$F_9(x) = f_9(M(x - o_9)) + F_9^* \quad (54)$$

#### CEC funkcija 10

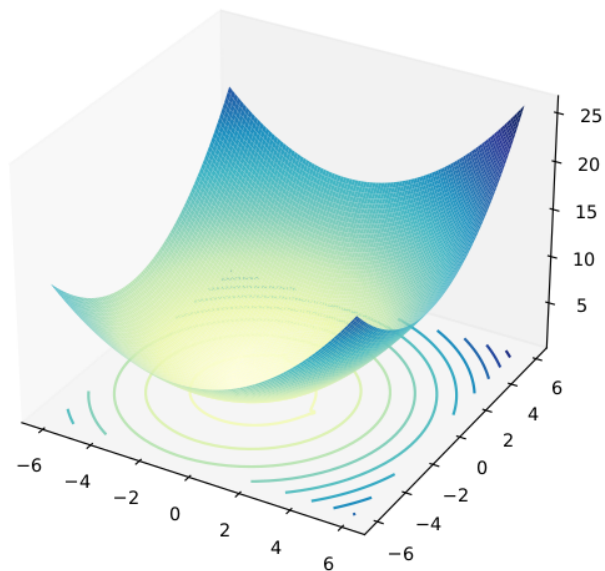
##### Ackley funkcija

$$f_{10}(\mathbf{x}) = -a \exp \left( -b \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2} \right) - \exp \left( \frac{1}{D} \sum_{i=1}^D \cos(c x_i) \right) + a + \exp(1) \quad (55)$$

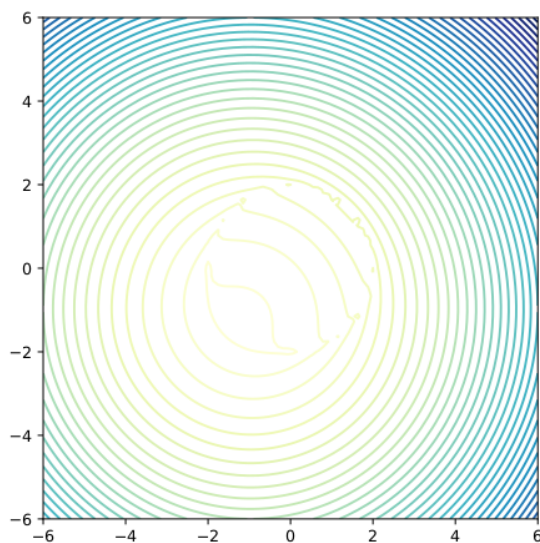
Preporučene vrednosti promenljivih su:  $a = 20$ ,  $b = 0,2$  i  $c = 2\pi$ .

##### Pomerena i rotirana Ackley funkcija

$$F_{10}(x) = f_{10}(M(x - o_{10})) + F_{10}^* \quad (56)$$



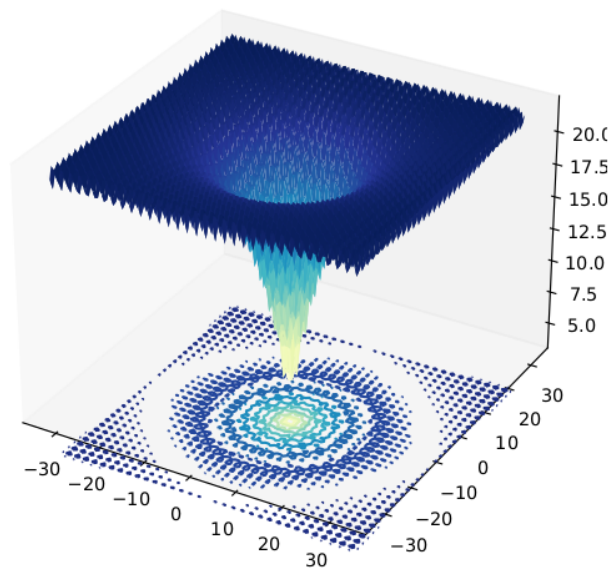
Slika 31: 3D grafikon dvodimenzionalne Happy Cat funkcije



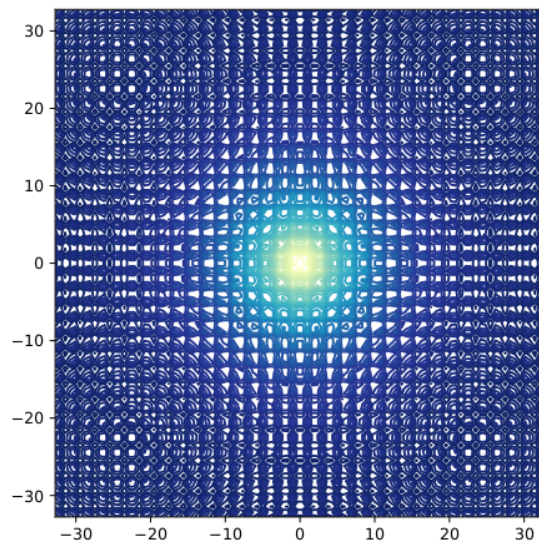
Slika 32: Kontur grafikon dvodimenzionalne Happy Cat funkcije

### 6.1.2 Rezultati na neograničenim funkcijama

U sprovedenim eksperimentima veličina populacije je 50, kriterijum za zastavljanje (broj maksimalnih iteracija) je 500. Simulacije su ponovljene u 30 nezavisnih



Slika 33: 3D grafikon dvodimenzionalne Ackley funkcije



Slika 34: Kontur grafikon dvodimenzionalne Ackley funkcije

pokretanja algoritma da bi dobijeni rezultati bili statistički validni. Pregled kontrolnih parametara je prikazan u tabeli 2.

Dobijeni statistički rezultati i upoređivanje prosečne vrednosti, standardne devijacije predložene HEFFF metode i uporedne metode su predstavljene u tabeli 3.

Tabela 2: Kontrolni parametri

| Opis parametara                     | Notacija   | Vrednost |
|-------------------------------------|------------|----------|
| Veličina populacije                 | $NP$       | 50       |
| Broj maksimalnih iteracija          | $MaxIter$  | 500      |
| Koeficijent apsorpcije              | $\gamma$   | 1        |
| Parametar atraktivnosti pri $r = 0$ | $\beta_0$  | 1        |
| Početna vrednost $\alpha$ value     | $\alpha_0$ | 0.5      |

Rezultati komparabilnih metoda <sup>1</sup> EHOI, EHO, SCA, SSA, GOA, WOA, BBO, MFO, i PSO su uzeti iz rada [154].

Tabela 3: Statistički rezultati simulacije na neograničenim CEC 2019 funkcijama

| Funkcija | Metrika | 7<br>EHOI             | Metoda                |                      |                       |                      |                      |                      |                       |                      |                      |                      |
|----------|---------|-----------------------|-----------------------|----------------------|-----------------------|----------------------|----------------------|----------------------|-----------------------|----------------------|----------------------|----------------------|
|          |         |                       | EHO                   | SCA                  | SSA                   | GOA                  | WOA                  | BBO                  | MFO                   | PSO                  | FFO                  | HEFF                 |
| CEC01    | prosek  | $4.7 \times 10^4$     | $1.4 \times 10^7$     | $9.5 \times 10^9$    | $2.7 \times 10^9$     | $1.5 \times 10^{10}$ | $1.1 \times 10^{10}$ | $3.4 \times 10^{10}$ | $6.6 \times 10^9$     | $8.7 \times 10^{11}$ | $1.5 \times 10^3$    | $6.8 \times 10^3$    |
|          | SD      | $2.9 \times 10^3$     | $7.8 \times 10^6$     | $7.9 \times 10^9$    | $2.6 \times 10^9$     | $3.1 \times 10^{10}$ | $8.7 \times 10^9$    | $2.6 \times 10^{10}$ | $8.5 \times 10^9$     | $9.2 \times 10^{11}$ | $4.5 \times 10^2$    | $2.1 \times 10^2$    |
| CEC02    | prosek  | $1.7 \times 10^1$     | $1.7 \times 10^1$     | $1.7 \times 10^1$    | $1.7 \times 10^1$     | $1.7 \times 10^1$    | $1.7 \times 10^1$    | $9.2 \times 10^1$    | $1.7 \times 10^1$     | $1.0 \times 10^4$    | $2.8 \times 10^0$    | $1.5 \times 10^0$    |
|          | SD      | $1.2 \times 10^{-15}$ | $4.6 \times 10^{-15}$ | $4.2 \times 10^{-2}$ | $8.0 \times 10^{-5}$  | $1.5 \times 10^{-2}$ | $2.8 \times 10^{-3}$ | $2.7 \times 10^1$    | $3.7 \times 10^{-15}$ | $3.8 \times 10^3$    | $1.4 \times 10^2$    | $9.6 \times 10^1$    |
| CEC03    | prosek  | $1.3 \times 10^1$     | $1.3 \times 10^1$     | $1.3 \times 10^1$    | $1.3 \times 10^1$     | $1.3 \times 10^1$    | $1.3 \times 10^1$    | $1.3 \times 10^1$    | $1.3 \times 10^1$     | $1.3 \times 10^1$    | $9.7 \times 10^0$    | $9.7 \times 10^0$    |
|          | SD      | $1.9 \times 10^{-15}$ | $1.9 \times 10^{-15}$ | $1.0 \times 10^{-4}$ | $2.4 \times 10^{-15}$ | $1.2 \times 10^{-4}$ | $1.4 \times 10^{-7}$ | $2.6 \times 10^{-7}$ | $3.5 \times 10^{-5}$  | $6.5 \times 10^{-4}$ | $4.3 \times 10^{-1}$ | $5.0 \times 10^{-1}$ |
| CEC04    | prosek  | $1.3 \times 10^1$     | $1.5 \times 10^1$     | $1.1 \times 10^3$    | $3.4 \times 10^1$     | $1.5 \times 10^2$    | $3.1 \times 10^2$    | $7.8 \times 10^1$    | $1.3 \times 10^2$     | $7.3 \times 10^1$    | $4.0 \times 10^0$    | $1.0 \times 10^0$    |
|          | SD      | $3.9 \times 10^0$     | $6.3 \times 10^0$     | $3.9 \times 10^2$    | $1.2 \times 10^1$     | $2.0 \times 10^2$    | $1.2 \times 10^2$    | $2.6 \times 10^1$    | $1.8 \times 10^2$     | $7.8 \times 10^0$    | $4.3 \times 10^{-1}$ | $1.8 \times 10^0$    |
| CEC05    | prosek  | $1.0 \times 10^0$     | $1.0 \times 10^0$     | $2.2 \times 10^0$    | $1.2 \times 10^0$     | $1.3 \times 10^0$    | $1.6 \times 10^0$    | $1.3 \times 10^0$    | $1.1 \times 10^0$     | $1.5 \times 10^0$    | $1.0 \times 10^0$    | $1.0 \times 10^0$    |
|          | SD      | $2.1 \times 10^{-2}$  | $2.2 \times 10^{-2}$  | $7.6 \times 10^{-2}$ | $1.1 \times 10^{-1}$  | $1.2 \times 10^{-1}$ | $4.0 \times 10^{-1}$ | $9.8 \times 10^{-2}$ | $8.0 \times 10^{-2}$  | $1.2 \times 10^{-1}$ | $2.4 \times 10^{-2}$ | $2.4 \times 10^{-2}$ |
| CEC06    | prosek  | $8.3 \times 10^0$     | $9.5 \times 10^0$     | $1.1 \times 10^1$    | $3.7 \times 10^0$     | $6.2 \times 10^0$    | $9.0 \times 10^0$    | $5.8 \times 10^0$    | $5.3 \times 10^0$     | $1.1 \times 10^1$    | $1.6 \times 10^0$    | $1.3 \times 10^0$    |
|          | SD      | $8.2 \times 10^{-1}$  | $1.3 \times 10^0$     | $7.4 \times 10^{-1}$ | $1.4 \times 10^0$     | $1.3 \times 10^0$    | $1.1 \times 10^0$    | $6.5 \times 10^{-1}$ | $2.2 \times 10^0$     | $6.7 \times 10^{-1}$ | $6.2 \times 10^{-2}$ | $4.6 \times 10^{-2}$ |
| CEC07    | prosek  | $1.4 \times 10^2$     | $1.8 \times 10^2$     | $6.6 \times 10^2$    | $2.9 \times 10^2$     | $3.0 \times 10^2$    | $4.5 \times 10^2$    | $4.8 \times 10^0$    | $3.2 \times 10^2$     | $6.1 \times 10^2$    | $5.9 \times 10^0$    | $3.9 \times 10^0$    |
|          | SD      | $1.0 \times 10^2$     | $1.5 \times 10^2$     | $1.5 \times 10^2$    | $2.3 \times 10^2$     | $1.7 \times 10^2$    | $2.2 \times 10^2$    | $1.3 \times 10^2$    | $2.1 \times 10^2$     | $1.6 \times 10^2$    | $1.1 \times 10^2$    | $9.2 \times 10^1$    |
| CEC08    | prosek  | $2.7 \times 10^0$     | $2.8 \times 10^0$     | $6.0 \times 10^0$    | $5.2 \times 10^0$     | $5.5 \times 10^0$    | $5.8 \times 10^0$    | $4.6 \times 10^0$    | $5.7 \times 10^0$     | $5.1 \times 10^0$    | $1.1 \times 10^0$    | $1.6 \times 10^0$    |
|          | SD      | $8.8 \times 10^{-1}$  | $1.1 \times 10^0$     | $5.4 \times 10^{-1}$ | $6.3 \times 10^{-1}$  | $8.0 \times 10^{-1}$ | $7.9 \times 10^{-1}$ | $1.1 \times 10^0$    | $5.8 \times 10^{-1}$  | $7.4 \times 10^{-1}$ | $8.0 \times 10^{-1}$ | $5.2 \times 10^{-1}$ |
| CEC09    | prosek  | $2.3 \times 10^0$     | $2.4 \times 10^0$     | $1.0 \times 10^2$    | $2.4 \times 10^0$     | $2.5 \times 10^0$    | $4.7 \times 10^0$    | $3.5 \times 10^0$    | $2.5 \times 10^0$     | $2.9 \times 10^0$    | $3.9 \times 10^0$    | $1.9 \times 10^0$    |
|          | SD      | $6.2 \times 10^{-3}$  | $1.3 \times 10^{-2}$  | $9.3 \times 10^1$    | $4.5 \times 10^{-2}$  | $7.2 \times 10^{-2}$ | $7.8 \times 10^{-1}$ | $2.3 \times 10^{-1}$ | $6.0 \times 10^{-2}$  | $9.7 \times 10^{-2}$ | $2.8 \times 10^{-1}$ | $1.0 \times 10^{-2}$ |
| CEC10    | prosek  | $2.0 \times 10^1$     | $2.0 \times 10^1$     | $2.0 \times 10^1$    | $2.0 \times 10^1$     | $2.0 \times 10^1$    | $2.0 \times 10^1$    | $2.0 \times 10^1$    | $2.0 \times 10^1$     | $2.0 \times 10^1$    | $2.1 \times 10^1$    | $1.1 \times 10^0$    |
|          | SD      | $1.5 \times 10^0$     | $9.8 \times 10^{-2}$  | $8.1 \times 10^{-2}$ | $8.3 \times 10^{-2}$  | $9.1 \times 10^{-2}$ | $4.9 \times 10^{-2}$ | $2.4 \times 10^{-2}$ | $1.5 \times 10^{-1}$  | $1.0 \times 10^{-1}$ | $9.1 \times 10^{-6}$ | $1.0 \times 10^{-2}$ |

Prezentovani rezultati u tabeli 3 ukazuju na superiornost HEFFF metode u odnosu na druge komparabilne metaheurističke metode. HEFFF algoritam ima najbolju prosečnu vrednost u slučaju 8 CEC test funkcija, dok originalni FFO ima najbolji prosečni rezultat kod dve funkcije, CEC01 i CEC08.

Pored upoređivanja prosečne vrednosti i standardne devijacije, neparametarski statistički testovi (eng. *nonparametric statistical tests*) su sprovedeni za rigoroznije upoređenje među metodama.

Kod inferencijalne statistike, testiranje hipoteza može se koristiti za izvođenje zaključaka o jednoj ili više populacija iz datatih uzoraka (rezultata). Da bismo to uradili, dve hipoteze su definisane, nulta hipoteza ( $H_0$ ) i alternativna hipoteza ( $H_1$ ). Nulta hipoteza je izjava o tome da nema razlike ili nema efekta, dok alternativna hipoteza predstavlja prisustvo efekta ili razlike, u ovoj studiji se odnosi na značajnu razliku između algoritama. Kada se primenjuje statistički postupak za odbacivanje hipoteze, nivo značaja  $\alpha$  se koristi za određivanje toga da na kom nivu hipoteza može biti odbačena.

Umesto da se odredi a priori nivo značaja  $\alpha$ , moguće je izračunati najmanji nivo značaja koji rezultira odbacivanjem nulte hipoteze. U statističkom testiranju, p-vrednost (vrednost verovatnoće) je verovatnoća da se dobije rezultat bar toliko ekstremno kao onaj koji je zapravo posmatrano, pod pretpostavkom da je  $H_0$  načn. Vrednost verovatnoće pruža informaciju o tome da li je statistički test hipoteze značajna ili ne, a takođe ukazuje na to koliko je rezultat signifikantan. Manja p-vrednost ukazuje na jak dokaz protiv  $H_0$ .

Kod statističke procedura za upoređenje u paru vrši se individualna poređenja između dva algoritma, dobijajući p-vrednost nezavisno od druge metode. Stoga je potrebno korišćenje testova višestrukih poređenja da bi se uključilo više algoritama u poređenje.

Od statističkih testova za upoređenje u paru, korišćen je Fridmanov test [161, 162], kao najpoznatiji postupak za ispitivanje razlika između više od dva povezana uzorka. Takođe je sprovedena dvosmerna analiza varijanse po rangovima (eng. *two-way analysis of variance by ranks*) u cilju dokazivanja značajne razlike između predloženog pristupa i drugih uporedivih metoda. Rangiranje CEC funkcija je predstavljeno u tabeli 4.

Tabela 4: Fridmanovo rangiranje uporedivih metoda na CEC 2019 test funkcijama

| Funkcija | Metoda |      |     |     |      |      |      |      |     |     |       |
|----------|--------|------|-----|-----|------|------|------|------|-----|-----|-------|
|          | EHOI   | EHO  | SCA | SSA | GOA  | WOA  | BBO  | MFO  | PSO | FFO | HEFFF |
| CEC01    | 3      | 4    | 7   | 5   | 9    | 8    | 10   | 6    | 11  | 1   | 2     |
| CEC02    | 5      | 5    | 9   | 5   | 8    | 5    | 10   | 5    | 11  | 2   | 1     |
| CEC03    | 7      | 7    | 7   | 7   | 7    | 7    | 7    | 7    | 7   | 2   | 1     |
| CEC04    | 3      | 4    | 11  | 5   | 9    | 10   | 7    | 8    | 6   | 2   | 1     |
| CEC05    | 3.5    | 3.5  | 11  | 6   | 8    | 10   | 7    | 5    | 9   | 2   | 1     |
| CEC06    | 7      | 9    | 11  | 3   | 6    | 8    | 5    | 4    | 10  | 2   | 1     |
| CEC07    | 4      | 5    | 11  | 6   | 7    | 9    | 3    | 8    | 10  | 2   | 1     |
| CEC08    | 3      | 4    | 11  | 7   | 8    | 10   | 5    | 9    | 6   | 1   | 2     |
| CEC09    | 2      | 3    | 11  | 4   | 5    | 10   | 8    | 6    | 7   | 9   | 1     |
| CEC10    | 2      | 8    | 10  | 3   | 4.5  | 6.5  | 4.5  | 6.5  | 9   | 11  | 1     |
| Prosek   | 3.95   | 5.25 | 9.9 | 5.1 | 7.15 | 8.35 | 6.65 | 6.45 | 8.6 | 3.4 | 1.2   |
| Rang     | 3      | 5    | 11  | 4   | 8    | 9    | 7    | 6    | 10  | 2   | 1     |

Tabela 4 prikazuje superiornost HEFFF-a u upoređivanju sa ostalih 10 algoritma sa prosečnim rangom od 1,2 kod Fridmanovog testa. Fridmanova statistika ( $\chi_r^2 = 58.9$ ) je veća od  $\chi^2$  kritične vrednosti sa 10 stepeni slobode (18.3), na nivou značaja  $\alpha = 0.005$ , prema tome nulta hipoteza ( $H_0$ ) je odbačena i možemo zaključiti da predloženi HEFFF značajno bolji od ostalih 10 metoda.

Iman-Davenportov test [163] može biti tačniji od aproksimacije hi-kvadratne vrednosti, kako je objavljeno u [164] i iz tog razloga, tokode je sproveden Iman-Davenportov test. Iman-Davenportov test je izvedeno iz Fridmanove statistike. Vrednost Iman-Davenportov testa se dobija sledećim izrazom:

$$F_{ID} = \frac{(n-1)\chi_F^2}{n(k-1) - \chi_F^2} \quad (57)$$

gde je  $n$  broj razmatranih problema, a  $k$  predstavlja broj algoritama uključenih u komparaciju.

Dobijena statistička vrednost Iman-Davenportov testa je  $1.29 \times 10$  koja je veća od distribucione kritične vrednosti  $F(F(9, 9 \times 10) = 1.94)$ , što ukazuje na postojanje značajnih razlika među algoritama koji su uzeti u obzir. Shodno tome, drugi statistički test takode odbija  $H_0$ . U oba testa,  $p$  vrednost je manja od nivoa značajnosti. Rezime statističkih rezultata je prikazano u tabeli 5.

Obzirom da i Fridmanov i Iman Davenportov testovi odbacuju nultu hipotezu, sprovedena je i neparametarska post-hoc procedura sa Holmovom (eng. *Holm's step-down*) procedurom. Primena post-hoc testa može dovesti do dobijanja  $p$ -vrednosti koja određuje stepen odbacivanja svake hipoteze.

Tabela 5: Fridmanov i Iman-Davenportov statistički test rezultati ( $\alpha = 0.05$ )

| Statistika                | Vrednost               |
|---------------------------|------------------------|
| Fridmanova vrednost       | $5.89 \times 10$       |
| $\chi^2$                  | $1.83 \times 10$       |
| Kritična vrednost         | $1.11 \times 10^{-16}$ |
| Iman-Davenportov vrednost | $1.29 \times 10$       |
| $F$ Kritična vrednost     | 1.94                   |
| $p$ -vrednost             | $1.11 \times 10^{-13}$ |

Dobijeni rezultati Holmove procedure su prikazani u tabeli 6.

Tabela 6: Rezultati Holmove procedure

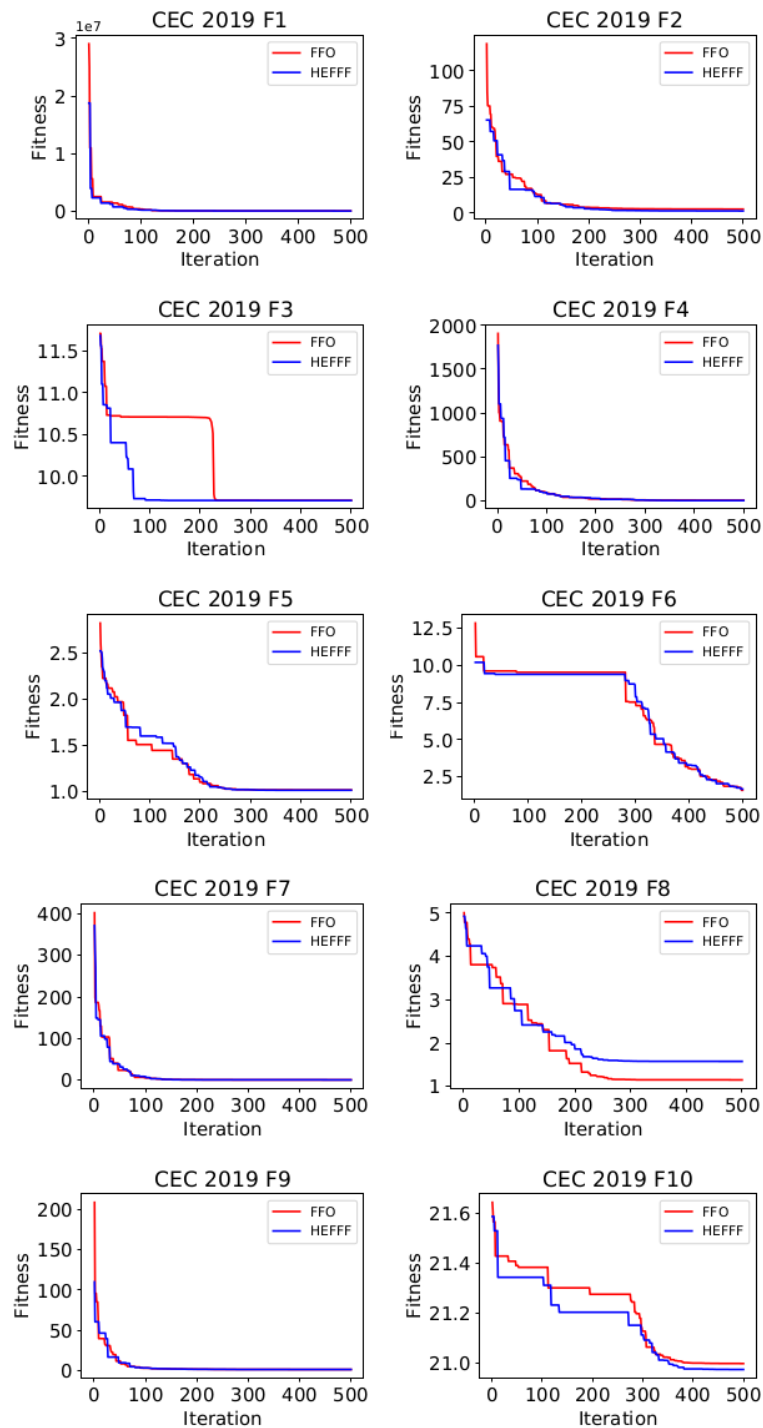
| Comparison | $p$ -Value            | Rank | 0.05/(k-i) | 0.1/(k-i) |
|------------|-----------------------|------|------------|-----------|
| SCA        | $2.24 \times 10^{-9}$ | 0    | 0.005000   | 0.01000   |
| PSO        | $3.03 \times 10^{-7}$ | 1    | 0.005556   | 0.01111   |
| WOA        | $7.16 \times 10^{-7}$ | 2    | 0.006250   | 0.01250   |
| GOA        | $3.02 \times 10^{-5}$ | 3    | 0.007143   | 0.01429   |
| BBO        | $1.1 \times 10^{-4}$  | 4    | 0.008333   | 0.01667   |
| MFO        | $2.00 \times 10^{-4}$ | 5    | 0.010000   | 0.02000   |
| EHO        | $3.16 \times 10^{-3}$ | 6    | 0.012500   | 0.02500   |
| SSA        | $4.28 \times 10^{-3}$ | 7    | 0.016667   | 0.03333   |
| EHOI       | $3.19 \times 10^{-2}$ | 8    | 0.025000   | 0.05000   |
| FFO        | $6.90 \times 10^{-2}$ | 9    | 0.050000   | 0.10000   |

Tabela 6 pokazuje da je predloženi metod značajno nadmašio uporedive metode na novou značajnosti  $\alpha = 0,1$ , kao i prvih osam metoda na nivou značajnosti  $\alpha = 0,05$ .

Slika 35 ilustruje direktno poređenje između HEFFF-a i druge najbolje metode, koji je FFO, na rezultatima 10 benčmark funkcija. Grafikoni ilustruju kako se vrednost ciljne funkcije smanjuje kroz 500 iteracija za svaku CEC test funkciju.

## 6.2 Eksperimenti klasterovanja tekstualnih dokumenata

U simulacijama klasterovanja tekstualnih dokumenata korišćeno je 6 različitih tekstualnih setova podataka za validaciju predložene HEFFF metode. Korišćeni podaci predstavljaju dobro poznate skupove podataka koji se koriste kao benčmark podaci u rudarenju tekstova i za evaulaciju metode klasterovanja nad tekstualnim podacima.



Slika 35: Grafikoni konvergencije HEFFF i FFO metode na CEC 2019 test funkcijama

### 6.2.1 Tekstualni skupovi podataka korišćeni u simulacijama klasterovanja tekstualnih dokumenata

<sup>8</sup> **Centre for Speech Technology Research (CSTR)** Skup podataka <sup>8</sup> *Centre for Speech Technology Research* se sastoji od 299 dokumenata. Dokumenti u skupu podataka pripadaju sledećim kategorijama:

- veštačka inteligencija,
- teorija,
- sistemi,
- robotika.

**20 Newsgroups** *20 Newsgroups* je kolekcija podataka koji se sastoji od 20.000 dokumenata, i dokumenti su grupisani 20 različitih kategorija. Ova kolekcija sadrži balansirani set podataka, dokumenti u klasama su ravnomerno raspoređeni.

U ovom istraživanju je korišćeno 3 grupa od ukupnih 20:

- <sup>1</sup> *the comp.windows.x*,
- talk.politics.misc*,
- rec.autos*.

Svaka od gore navedenih 3 kategorija sadrži 100 dokumenata.

**Tr41** *Tr41* korpus je iz Karypis laboratorije, skup podataka sadrži 10 klasa. Podaci su iz *Text REtrieval Conference* (TREC) projekta <sup>1</sup> [165].

**Tr12** *Tr12* korpus je iz Karypis laboratorije, skup podataka sadrži 8 klasa. Podaci su iz *Text REtrieval Conference* (TREC) projekta <sup>2</sup> [166].

**Wap** *Wap* korpus je iz Karypis laboratorije, skup podataka sadrži 20 klasa. Podaci su iz projekta WebACE [167] gde svaki dokument odgovara veb stranici navedenoj u predmetnoj hijerarhiji *Yahoo!*.

---

<sup>1</sup>www.trec.nist.gov

<sup>2</sup>www.trec.nist.gov

**Classic 4** Skup podataka *Classic 4* je balansiran skup podataka koji se sastoji od podataka sažime radova i podeljene su u 4 klase, svaka klasa ima 500 dokumenata, korpus sadrži ukupno 2.000 dokumenata.

- CISI,
- CACM,
- MED,
- CRAN.

Rezime karakteristika skupa tekstualnih podataka se nalazi u tabeli 7.

Tabela 7: Karakteristika skupa tekstualnih podataka korišćene za simulacije klasterovanja tekstualnih dokumenata

| Naziv         | Broj dokumenata | Broj klastera | Broj atributa<br>(pojмова) |
|---------------|-----------------|---------------|----------------------------|
| <b>1</b> CSTR | 299             | 4             | 1725                       |
| 20 Newsgroups | 300             | 3             | 2275                       |
| Tr12          | 313             | 8             | 5329                       |
| Tr41          | 878             | 10            | 6743                       |
| Wap           | 1560            | 20            | 7512                       |
| Classic4      | 2000            | 4             | 6500                       |

### 6.2.2 Mere evaulacije kvaliteta klasterovanja

Za procenu kvaliteta klasterovanja korišćene su sledeće konvencionalne metrike za evaulaciju performansi klasterovanja tekstualnih dokumenata:

- stopa greške (eng. *error-rate*),
- tačnost (eng. *accuracy*),
- preciznost (eng. *precision*),
- odziv (eng. *recall*),
- F-mera (eng. *F-measure*),
- čistoća (eng. *purity*),
- entropija (eng. *entropy*).

**Stopa greške** Stopa greške predstavlja osnovnu meru za evaulaciju performansi kreiranih modela. Stopa greške predstavlja procenat pogrešno dodeljenih klastera dokumentima u datom skupu podataka, veći procenat ukazuje na nestabilan model, i obrnuto, što je procenat manji, model je bolji. Matematička formulacija je prikazana u jednačini (58):

$$stopa\_greške = \frac{1}{n} \sum_{j=1}^k n_{i,j}^- \quad (58)$$

gde ukupan broj dokumenata u skupu podataka označen sa  $n$ ,  $n_{i,j}^-$  predstavlja broj pogrešno dodeljenih klastera dokumentima iz klase  $i$  u klaster  $j$ , a  $k$  je broj klastera.

**Tačnost** Mera tačnosti predstavlja odnos tačno dodeljenih klastera u datom skupu podataka. Vrednost se dobija preko sledećeg izraza:

$$tačnost = \frac{1}{n} \sum_{j=1}^k n_{i,j} \quad (59)$$

gde  $n$  predstavlja ukupan broj dokumenata u skupu podataka,  $n_{i,j}$  predstavlja broj tačno dodeljenih klastera dokumentima iz klase  $i$  u klaster  $j$ , a  $k$  je broj klastera.

**Preciznost** Metrika preciznost daje informacije o tačnim klasama za sve klase u klasterima, a izračunava se pomoću izraza (60):

$$P(i, j) = \frac{n_{i,j}}{n_j} \quad (60)$$

gde ispravno dodeljene klase  $i$  klsteru  $j$  je označeno sa  $n_{i,j}$ , a  $n_j$  označava ukupan broj dokumenata u klasteru  $j$ .

**Odziv** Odziv predstavlja odnos tačno dodeljenih dokumenata u odnosu na ukupan broj dokumenata u datoj klasi i vrednost se računa jednačinom (61):

$$R(i, j) = \frac{n_{i,j}}{n_i} \quad (61)$$

gde  $n_{i,j}$  označava ispravno dodeljenu klasu  $i$  klasteru  $j$ . Ukupan broj dokumenata u klasi  $i$  je predstavljeno sa  $n_i$ .

**F-mera** Vrednost F-mera predstavlja harmoničnu sredinu preciznosti i doziva i vrednost se računa preko izraza (62):

$$F(i, j) = \frac{2 \times P(i, j) \times R(i, j)}{P(i, j) + R(i, j)} \quad (62)$$

gde  $P(i, j)$  i  $R(i, j)$  označava preciznost  $i$  odziv, respektivno.

**Entropija** Entropija meri distribuciju dokumenata sa oznakama klasa u svakom klasteru i njegova vrednost je između 0 i 1, najbolja vrednost je 0 ili blizu 0. Vrednost entropije klastera  $j$  je određena jednačinom (63):

$$E(j) = - \sum_{i=1}^{17} p(i, j) \log p(i, j) \quad (63)$$

gde  $E(j)$  označava entropiju klastera  $j$ , a  $p(i, j)$  predstavlja verovatnoću klase  $i$  u klasteru  $j$ .

Entropija svih klastera je određena prema sledećem izrazu:

$$E = \sum_{j=1}^k \frac{n_j}{n} E(j) \quad (64)$$

### 6.2.3 Rezultati i analiza eksperimenta klasterovanja

Eksperimenti klasterovanja tekstualnih dokumenata izvršeno u 30 nezavisnih pokretanja algoritma da bi dobijeni rezultati bili statistički validni. Maksimalan broj iteracije je postavljeno na 1000, broj rešenja u populaciji je 20. Rezime inicijalnih kontrolnih parametara je predstavljeno u tabeli 8.

Tabela 8: Inicijalni vrednosti kontrolnih parametara

| Naziv parametara                    | Notacija   | Vrednost |
|-------------------------------------|------------|----------|
| Veličina populacije                 | $NP$       | 20       |
| Broj maksimalnih iteracija          | $MaxIter$  | 1000     |
| Koeficijent apsorpcije              | $\gamma$   | 1        |
| Parametar atraktivnosti pri $r = 0$ | $\beta_0$  | 1        |
| Početna vrednost $\alpha$ value     | $\alpha_0$ | 0.5      |

Rezultati predložene HEFFF metode na eksperimentima klasterovanja tekstualnih dokumenata na ranije opisanim skupovima podataka, kao i upoređivanje sa original-

nim FFO metodom i dvevet drugih metaheurističkih metoda i pet nemetaheurističkih metoda su predstavljeni u tabelama 9–14.

Rezultati komparativnih metoda (K-mean, K-mean++, DBSCAN, Agglomerative, Spectral, KHA, HS, PSO, GA, MVO, H-PSO, H-GA, H-MVO1, H-MVO2) su uzeti iz rada [128].

Tabela 9: Rezultati klasterovanja tekstualnih podataka na CSTR skupu podataka

| Metoda        | Stopa greške | Tačnost | Preciznost | Odziv  | F-mera | Čistoća | Entropija |
|---------------|--------------|---------|------------|--------|--------|---------|-----------|
| K-mean        | 0.4029       | 0.3573  | 0.4091     | 0.3092 | 0.346  | 0.3525  | 0.8201    |
| K-mean++      | 0.3678       | 0.4355  | 0.3953     | 0.4076 | 0.3546 | 0.4096  | 0.5246    |
| DBSCAN        | 0.4244       | 0.4005  | 0.3393     | 0.4256 | 0.3046 | 0.4076  | 0.4586    |
| Agglomerative | 0.3672       | 0.436   | 0.4423     | 0.4666 | 0.3266 | 0.4816  | 0.5076    |
| Spectral      | 0.3591       | 0.4319  | 0.3597     | 0.4925 | 0.3971 | 0.4485  | 0.4893    |
| KHA           | 0.3328       | 0.3649  | 0.4213     | 0.5355 | 0.4139 | 0.3874  | 0.4344    |
| HS            | 0.3431       | 0.4464  | 0.4235     | 0.506  | 0.3377 | 0.4355  | 0.4786    |
| PSO           | 0.3374       | 0.4356  | 0.534      | 0.436  | 0.4819 | 0.4953  | 0.6199    |
| GA            | 0.3948       | 0.3399  | 0.4417     | 0.3418 | 0.3886 | 0.405   | 0.717     |
| MVO           | 0.3198       | 0.4593  | 0.5715     | 0.4829 | 0.5244 | 0.5685  | 0.5207    |
| H-PSO         | 0.3028       | 0.5494  | 0.6065     | 0.56   | 0.5577 | 0.6135  | 0.5076    |
| H-GA          | 0.3078       | 0.5056  | 0.613      | 0.495  | 0.5859 | 0.5743  | 0.5104    |
| H-MVO1        | 0.2678       | 0.5683  | 0.6395     | 0.5079 | 0.5774 | 0.5015  | 0.4577    |
| H-MVO2        | 0.2358       | 0.5779  | 0.6497     | 0.5568 | 0.5936 | 0.598   | 0.401     |
| FFO           | 0.2285       | 0.5859  | 0.5757     | 0.5944 | 0.5849 | 0.586   | 0.3397    |
| HEFFF         | 0.2215       | 0.5964  | 0.5873     | 0.5641 | 0.5755 | 0.6140  | 0.3361    |

Tabela 10: Rezultati klasterovanja tekstualnih podataka na 20 Newsgroups skupu podataka

| Metoda        | Stopa greške | Tačnost | Preciznost | Odziv  | F-mera | Čistoća | Entropija |
|---------------|--------------|---------|------------|--------|--------|---------|-----------|
| K-mean        | 0.3205       | 0.318   | 0.3121     | 0.31   | 0.3406 | 0.3741  | 0.8028    |
| K-mean++      | 0.3257       | 0.3784  | 0.3652     | 0.3662 | 0.3619 | 0.4134  | 0.6611    |
| DBSCAN        | 0.384        | 0.3038  | 0.3094     | 0.3017 | 0.3193 | 0.3027  | 0.7473    |
| Agglomerative | 0.3313       | 0.4055  | 0.399      | 0.3576 | 0.3548 | 0.4417  | 0.5953    |
| Spectral      | 0.3143       | 0.3633  | 0.3424     | 0.328  | 0.3136 | 0.311   | 0.6125    |
| KHA           | 0.2962       | 0.3216  | 0.3829     | 0.3136 | 0.2996 | 0.3421  | 0.6767    |
| HS            | 0.3021       | 0.3122  | 0.3601     | 0.317  | 0.3214 | 0.3355  | 0.6481    |
| PSO           | 0.2967       | 0.3498  | 0.4134     | 0.3497 | 0.3803 | 0.4097  | 0.7723    |
| GA            | 0.3605       | 0.3676  | 0.4209     | 0.3676 | 0.3936 | 0.4081  | 0.7547    |
| MVO           | 0.2804       | 0.4044  | 0.4392     | 0.3842 | 0.4109 | 0.4344  | 0.7121    |
| H-PSO         | 0.2744       | 0.4442  | 0.4821     | 0.422  | 0.4454 | 0.4725  | 0.6631    |
| H-GA          | 0.2804       | 0.4308  | 0.4814     | 0.4887 | 0.4433 | 0.4837  | 0.6833    |
| H-MVO1        | 0.2864       | 0.5174  | 0.5102     | 0.4332 | 0.5169 | 0.5104  | 0.5811    |
| H-MVO2        | 0.2324       | 0.5326  | 0.5619     | 0.4876 | 0.5376 | 0.4591  | 0.5407    |
| FFO           | 0.2272       | 0.55    | 0.5376     | 0.55   | 0.5437 | 0.5100  | 0.3299    |
| HEFFF         | 0.2238       | 0.5833  | 0.5668     | 0.5833 | 0.575  | 0.5100  | 0.3267    |

Kao što je predstavljeno u tabelama, predložena hibridna metoda HEFFF je veoma obećavajuća i kompetentna metoda za klasterovanje tekstualnih dokumenata. HEFFF ima najbolju stopu greške u eksperimentu kod svih šest skupova podataka, a u slučaju pet skupova podataka ima najbolje vrednosti za odziv, dok F-mera najbolja kod četiri skupa podataka.

Na CSTR skupu podataka, HEFFF je značajno poboljšao tačnost, za 16% u

Tabela 11: Rezultati klasterovanja tekstualnih podataka na Tr12 skupu podataka

| Metoda        | Stopa greške  | Tačnost       | Preciznost    | Odziv         | F-mera       | Čistoća       | Entropija     |
|---------------|---------------|---------------|---------------|---------------|--------------|---------------|---------------|
| K-mean        | 0.4384        | 0.2971        | 0.3522        | 0.2944        | 0.3222       | 0.3908        | 0.7138        |
| K-mean++      | 0.4899        | 0.3795        | 0.4215        | 0.3778        | 0.4176       | 0.4808        | 0.5094        |
| DBSCAN        | 0.4271        | 0.3045        | 0.3218        | 0.3234        | 0.4048       | 0.4728        | 0.501         |
| Agglomerative | 0.4422        | 0.4481        | 0.4923        | 0.4341        | 0.409        | 0.5553        | 0.3978        |
| Spectral      | 0.4682        | 0.3373        | 0.4508        | 0.3055        | 0.3781       | 0.4132        | 0.4932        |
| KHA           | 0.4136        | 0.3357        | 0.3748        | 0.3099        | 0.2916       | 0.3803        | 0.5183        |
| HS            | 0.4484        | 0.3776        | 0.4385        | 0.3453        | 0.447        | 0.4986        | 0.4517        |
| PSO           | 0.4272        | 0.4075        | 0.4298        | 0.4264        | 0.4278       | 0.4878        | 0.572         |
| GA            | 0.4767        | 0.3677        | 0.4128        | 0.3549        | 0.3826       | 0.4513        | 0.6233        |
| MVO           | 0.4669        | 0.4485        | 0.5075        | 0.4398        | 0.4706       | 0.5448        | 0.5224        |
| H-PSO         | 0.4599        | 0.4796        | 0.5355        | 0.3893        | 0.586        | 0.5876        | 0.5757        |
| H-GA          | 0.4549        | 0.5205        | 0.4798        | 0.4974        | 0.5428       | <b>0.6038</b> | 0.638         |
| H-MVO1        | 0.4019        | 0.5465        | 0.6025        | 0.5138        | 0.5776       | 0.5948        | 0.4224        |
| H-MVO2        | 0.4279        | 0.5617        | <b>0.6398</b> | 0.5109        | 0.5956       | 0.6063        | <b>0.3753</b> |
| FFO           | 0.3859        | 0.5769        | 0.5992        | 0.5771        | 0.5879       | 0.535         | 0.6663        |
| HEFFF         | <b>0.3685</b> | <b>0.6398</b> | 0.6295        | <b>0.6284</b> | <b>0.629</b> | 0.5744        | 0.6965        |

Tabela 12: Rezultati klasterovanja tekstualnih podataka na Tr41 skupu podataka

| Metoda        | Stopa greške  | Tačnost       | Preciznost    | Odziv        | F-mera       | Čistoća       | Entropija    |
|---------------|---------------|---------------|---------------|--------------|--------------|---------------|--------------|
| K-mean        | 0.4459        | 0.4126        | 0.3945        | 0.3813       | 0.3876       | 0.4108        | 0.5874       |
| K-mean++      | 0.4816        | 0.426         | 0.3769        | 0.3559       | 0.4299       | 0.5471        | 0.4745       |
| DBSCAN        | 0.4552        | 0.3704        | 0.4242        | 0.4291       | 0.4152       | 0.6025        | 0.4776       |
| Agglomerative | 0.4405        | 0.461         | 0.3445        | 0.3487       | 0.4187       | 0.476         | 0.4821       |
| Spectral      | 0.4556        | 0.3787        | 0.363         | 0.348        | 0.3691       | 0.4866        | 0.5386       |
| KHA           | 0.4212        | 0.4054        | 0.4191        | 0.4579       | 0.4137       | 0.6063        | 0.4041       |
| HS            | 0.3885        | 0.359         | 0.371         | 0.3888       | 0.3701       | 0.4863        | 0.4539       |
| PSO           | 0.4185        | 0.487         | 0.4505        | 0.4497       | 0.4497       | 0.579         | 0.5391       |
| GA            | 0.5048        | 0.432         | 0.414         | 0.4008       | 0.4071       | 0.5603        | 0.5469       |
| MVO           | 0.4749        | 0.463         | 0.4569        | 0.4419       | 0.4569       | 0.6081        | 0.5355       |
| H-PSO         | 0.4389        | 0.474         | 0.501         | 0.4768       | 0.4701       | 0.5273        | 0.5029       |
| H-GA          | 0.4449        | 0.594         | 0.5245        | 0.5127       | 0.5647       | 0.642         | 0.5371       |
| H-MVO1        | 0.4029        | 0.568         | 0.5489        | 0.5079       | 0.5119       | 0.6421        | 0.3965       |
| H-MVO2        | 0.3939        | 0.615         | 0.569         | 0.5178       | 0.5201       | <b>0.6673</b> | 0.3669       |
| FFO           | 0.3852        | 0.617         | 0.6111        | 0.6132       | 0.6122       | 0.5475        | 0.316        |
| HEFFF         | <b>0.3769</b> | <b>0.6487</b> | <b>0.6341</b> | <b>0.642</b> | <b>0.638</b> | 0.5791        | <b>0.313</b> |

Tabela 13: Rezultati klasterovanja tekstualnih podataka na Wap skupu podataka

| Metoda        | Stopa greške  | Tačnost       | Preciznost    | Odziv         | F-mera       | Čistoća       | Entropija     |
|---------------|---------------|---------------|---------------|---------------|--------------|---------------|---------------|
| K-mean        | 0.4885        | 0.5012        | 0.4626        | 0.4011        | 0.4315       | 0.4759        | 0.7044        |
| K-mean++      | 0.5145        | 0.4937        | 0.4913        | 0.3508        | 0.4462       | 0.5797        | 0.5961        |
| DBSCAN        | 0.5372        | 0.4846        | 0.5004        | 0.4176        | 0.4878       | 0.4007        | 0.5706        |
| Agglomerative | 0.425         | 0.4933        | 0.4479        | 0.4471        | 0.4487       | 0.5445        | 0.4875        |
| Spectral      | 0.481         | 0.4905        | 0.4454        | 0.3423        | 0.4141       | 0.5142        | 0.5468        |
| KHA           | 0.4505        | 0.5191        | 0.4773        | 0.4106        | 0.4131       | 0.5939        | 0.5595        |
| HS            | 0.4725        | 0.5032        | 0.463         | 0.4447        | 0.4611       | 0.5589        | 0.547         |
| PSO           | 0.4598        | 0.5623        | 0.5249        | 0.4811        | 0.5017       | 0.6125        | 0.5765        |
| GA            | 0.5083        | 0.5316        | 0.5314        | 0.4706        | 0.4998       | 0.4917        | 0.6216        |
| MVO           | 0.4392        | 0.5291        | 0.5213        | 0.4496        | 0.4831       | 0.6069        | 0.6625        |
| H-PSO         | 0.4132        | 0.5532        | 0.545         | 0.4907        | 0.5521       | 0.6939        | 0.631         |
| H-GA          | 0.4272        | 0.6523        | 0.5749        | 0.5671        | 0.6187       | <b>0.7315</b> | 0.6585        |
| H-MVO1        | 0.3922        | 0.5931        | 0.6063        | 0.5906        | 0.5911       | 0.6849        | 0.4365        |
| H-MVO2        | 0.3762        | 0.6176        | 0.6344        | 0.5855        | 0.6105       | 0.7107        | <b>0.4196</b> |
| FFO           | 0.3859        | 0.5769        | 0.5992        | 0.5771        | 0.5879       | 0.535         | 0.6663        |
| HEFFF         | <b>0.3685</b> | <b>0.6398</b> | <b>0.6295</b> | <b>0.6284</b> | <b>0.629</b> | 0.5744        | 0.6965        |

odnosu na nemetaheurističke pristupe, a za 2% u odnosu na najbolje rezultiran metaheuristički metod prijavljenim u [128].

Kod eksperimenata na skupu podatka *20 Newsgroups*, HEFFF ima najbolje

Tabela 14: Rezultati klasterovanja tekstualnih podataka na Classic 4 skupu podataka

| Metoda          | Stopa greške  | Tačnost       | Preciznost    | Odziv         | F-mera        | Čistoća      | Entropija    |
|-----------------|---------------|---------------|---------------|---------------|---------------|--------------|--------------|
| <b>2</b> K-mean | 0.3461        | 0.5858        | 0.5699        | 0.5259        | 0.5472        | 0.5938       | 0.5601       |
| K-mean++        | 0.3735        | 0.6799        | 0.688         | 0.7028        | 0.692         | 0.6227       | 0.354        |
| DBSCAN          | 0.3951        | 0.6206        | 0.5708        | 0.5796        | 0.6173        | 0.6089       | 0.4854       |
| Agglomerative   | 0.3486        | 0.6683        | 0.6449        | 0.6224        | 0.6281        | 0.5812       | 0.4033       |
| Spectral        | 0.3348        | 0.6326        | 0.662         | 0.6528        | 0.6764        | 0.6352       | 0.4002       |
| KHA             | 0.3031        | 0.5761        | 0.6246        | 0.555         | 0.5529        | 0.555        | 0.5321       |
| HS              | 0.3166        | 0.5963        | 0.6064        | 0.5674        | 0.5787        | 0.5983       | 0.4946       |
| PSO             | 0.315         | 0.6363        | 0.6604        | 0.6164        | 0.6377        | 0.6243       | 0.5306       |
| GA              | 0.3694        | 0.6621        | 0.6726        | 0.632         | 0.6519        | 0.632        | 0.5781       |
| MVO             | 0.2951        | 0.7043        | 0.6919        | 0.6844        | 0.6881        | 0.6742       | 0.5113       |
| H-PSO           | 0.2861        | 0.6853        | 0.7354        | 0.6754        | 0.6297        | 0.6413       | 0.5656       |
| H-GA            | 0.2791        | 0.7273        | <b>0.7774</b> | 0.6944        | 0.6887        | 0.6733       | 0.6166       |
| H-MVO1          | 0.2461        | 0.7163        | 0.7129        | 0.6844        | 0.6951        | 0.7042       | 0.3553       |
| H-MVO2          | 0.2391        | 0.7271        | 0.7226        | 0.705         | 0.7079        | <b>0.732</b> | 0.3341       |
| FFO             | 0.2285        | 0.7296        | 0.6102        | 0.7236        | 0.6621        | 0.5501       | <b>0.262</b> |
| HEFFF           | <b>0.2218</b> | <b>0.7459</b> | 0.6622        | <b>0.7467</b> | <b>0.7029</b> | 0.6311       | 0.288        |

rezultate stope greške, tačnosti, preciznosti, odziva, F-mere i entropije, dok čistoća na istom skupu podataka je najbolja sa metodom H-MVO1.

Predloženi HEFFF ima najbolje performanse na *Tr12* skupu podataka, gde od statističkih mera ima najbolje vrednosti za stopu greške, tačnost, odziv i F-meru, dok je drugi najbolji pristup H-MVO2, koji je postigao najbolji rezultat u pogledu preciznosti, čistoće i entropije.

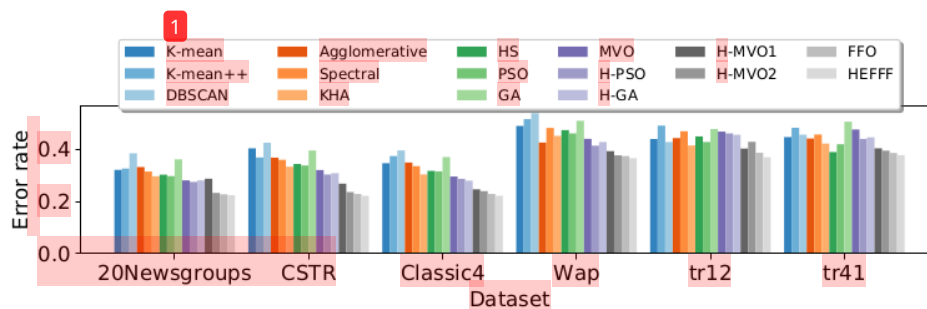
Poboljšanje tačnosti HEFFF-a kod *Tr41* skupa podataka je od 3% do 24% u odnosu na druge metode. U slučaju *Tr41*, HEFFF ima najbolji rezultat po svim metrikama, osim čistoće, gde najbolje rezultirajući metod je H-MVO2.

Na *Wap* skupu podataka, H-GA metod ima najbolju tačnost, dok je predložena metoda postigla najbolju vrednost za tri metrike.

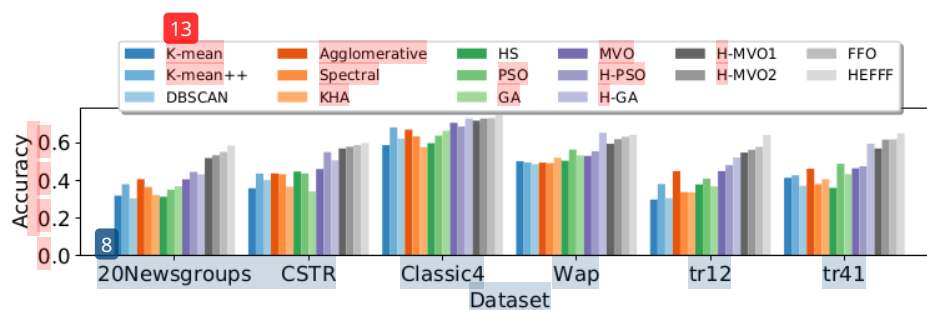
Na skupu podataka *Classic 4*, HEFFF je poboljšao tačnost sa 2% na 16% u odnosu na druge metaheurističke i nemetaheurističke pristupe.

Sveukupno, HEFFF je klasifikovan na prvom mestu u svih šest referentnih slučajeva po stopi greške i tačnosti. U 5 od 6 skupova podataka, predložena metoda ima najbolji odziv i F-meru. Jedini skup podataka gde HEFFF nije bio najbolji za odziv i F-meru je *CSTR*, ali dobijeni rezultati i kod ove dve mere su veoma blizu rezultatima nabolje metode na ovom skupu podataka. Najbolja preciznost i entropija se dobijaju kod 3 od 6 setova podataka, dok najbolja čistoća postiže u jednom referentnom slučaju.

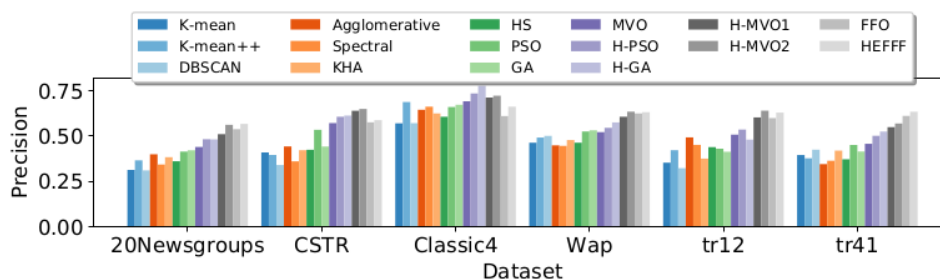
Slike 36–42 ilustruju komparaciju rezultata metrike HEFFF i ostalih metaheurističkih i nemetaheurističkih komparativnih metoda.



Slika 36: Uporedni prikaz rezultata stope greške HEFFF metode i ostalih komparabilnih metoda

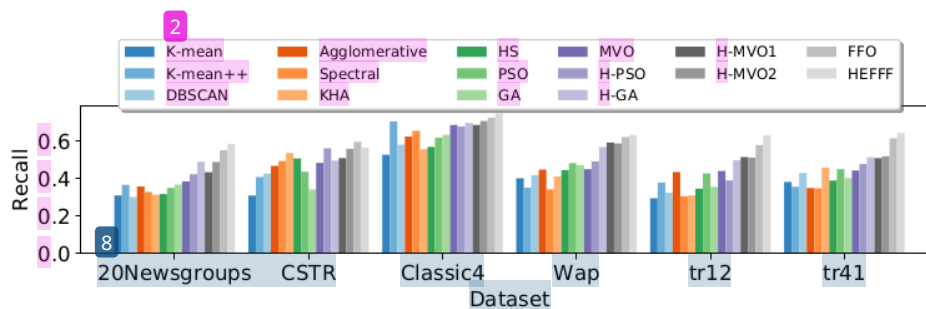


Slika 37: Uporedni prikaz rezultata tačnosti HEFFF metode i ostalih komparabilnih metoda

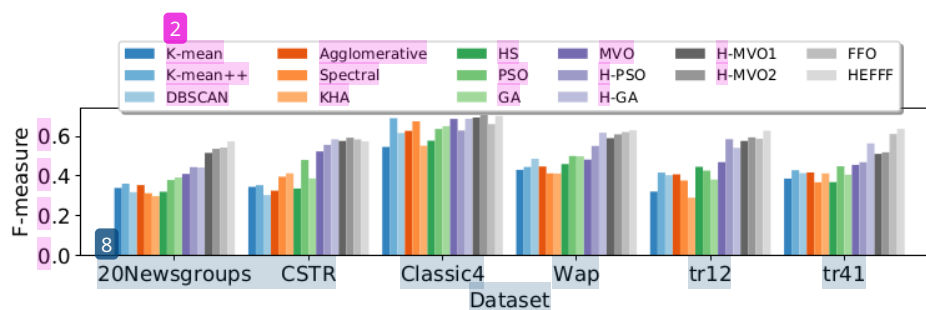


Slika 38: Uporedni prikaz rezultata preciznosti HEFFF metode i ostalih komparabilnih metoda

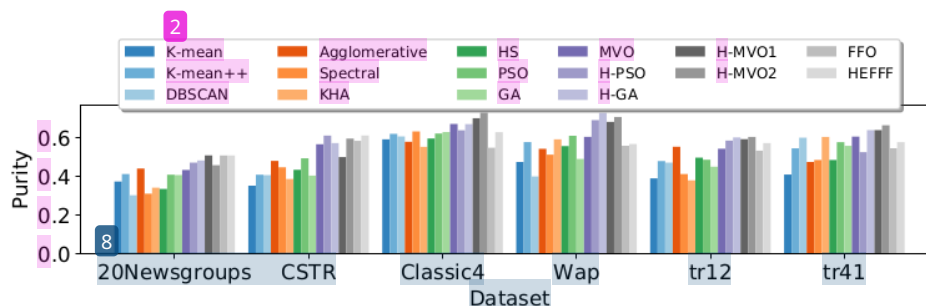
U eksperimentima sa neograničenim CEC funkcijama i u eksperimentima klasterovanja tekstulanih dokumenata se vrši poređenje između HEFFF, originalni FFO i drugih metaheuristika. Svi metaheuristički metodi su testirani pod sličnim uslovima u smislu broja iteracija i broja rešenja u populaciji. Zadržane su relativno niske vrednosti za veličinu populacije, koja je 50 kod eksperimenata sa CEC funkcijama i



Slika 39: Uporedni prikaz rezultata odziva HEFFF metode i ostalih komparabilnih metoda



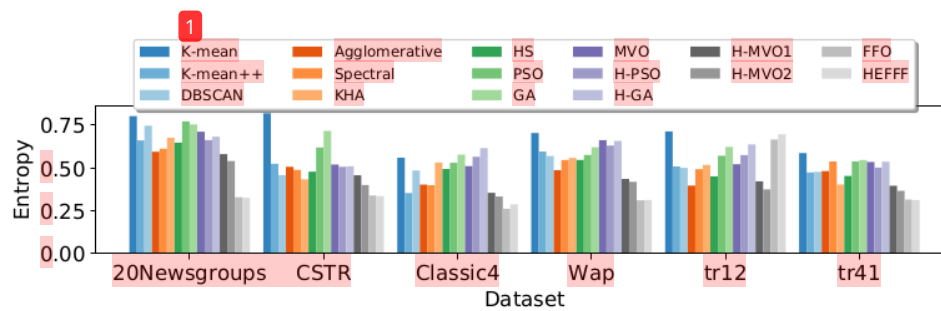
Slika 40: Uporedni prikaz rezultata F-mere HEFFF metode i ostalih komparabilnih metoda



Slika 41: Uporedni prikaz rezultata čistoće HEFFF metode i ostalih komparabilnih metoda

20 kod klasterovanje tekstualnih dokumenata, a broj maksimalnih iteracija je 500 i 1000, respektivno.

Na osnovu eksperimentalnih rezultata možemo zaključiti da je HEFFF veoma efikasna i robusna metoda u klasterovanju tekstualnih dokumenata. Faza FFO pretrage i OBL mehanizam istražuju prostor za pretragu na globalnom nivu, a



Slika 42: Uporedni prikaz rezultata entropije HEFFF metode i ostalih komparabilnih metoda

uključivanjem FA, eksploatacija ja poboljšana i pojačana, koji dovodi do efikasnog pronalaženja boljih centroida klastera od drugih komparabilnih pristupa.

## 7 Zaključak

Problem klasterovanje tekstualnih dokumenata je otvoreno pitanje za istraživače u okviru oblasti rudarenje teksta. U literaturi je predloženo više različitih metaheurističkih i nemetaheurističkih metoda za problem klasterovanje tekstualnih dokumenata, međutim i dalje postoji potreba da se razvije algoritam koji je robusniji, efikasniji od postojećih i koja prevazilazi nedostatke dosadašnjih tehnika. Glavni nedostatak algoritma K-srednjih vrednosti je inicijalizacija početnih vrednosti centroida i zaglavljivanje u lokalnom optimumu, dok je glavni nedostatak postojećih rešenja optimalni balans algoritma u fazama eksploracije i eksploatacije.

Ultimativni cilj ove disertacije je procena algoritama inteligencije roja i otkrivanje način poboljšanja performansi hibridizacijom komponenata iz više algoritma koji za rezultat ima savladanje nedostataka postojećih rešenja kod problema klasterovanja tekstualnih dokumenata. Dakle, cilj algoritma je da se inicijalno koristi algoritam inteligencije roja u cilju izbegavanja zaglavljivanja u optimumu i efikasne inicijalizacije i optimizacije centroida klastera koji nudi rešenje za problem koji ima algoritam K-srednjih vrednosti. Fuzija kvalitetnih komponenata različitih metoda inteligencije roja dovodi do dostizanja pravog balansa između eksploracije i eksploatacije. Na kraju algoritma se primenjuje K-srednjih vrednosti za finalno određivanje vrednosti centroida klastera. Na ovaj način se dostiže odlična separacija klastera, gde dokumenti u istim klasterima imaju visoku sličnost u smislu sadržaja, dok različiti klasteri imaju vrlo nisku sličnost.

Klasterovanje može biti formulisana kao problem optimizacije, gde je potrebno optimizovati centroe klastera tako što će funkcija koštanja biti minimizovana i ovaj problem pripada NP-teškim problemima. Pokazalo se da su algoritmi inteligencije roja veoma efikasni u rešavanju NP-teških problema.

U ovoj disertaciji je predložen hibridni metaheuristički algoritam koji je dodatno kombinovan sa algoritmom K-srednjih vrednosti za rešavanje problema klasterovanja tekstualnih dokumenata. Metoda je nazvana kao hibridna poboljšana voćna mušica svitac, skraćeno HEFFF, od engleskog naziva *hybrid enhanced fruit-fly firefly*. U predloženoj tehnici, kombinovane su prednosti dva algoritma inteligencije roja, FFO algoritam je hibridizovan sa FA i dodatno je ugrađen OBL mehanizam za bolje i efikasnije istraživanje prostora pretrage na globalnom nivou.

Kao standardna procedura provere performanse i dokazivanja efikasnosti novih metaheurističkih algoritama, koriste se benčmark test funkcije. Metodi koji pokazuju i verifikuju uspešnost u numeričkoj optimizaciji, smatraju se prikladnim za rešavanje problema iz realnog sveta. Stoga, je HEFFF prvo testiran na deset neograničenih benčmark CEC 2019 funkcija, zatim je primenjen na problem klasterovanja tekstualnih dokumenata.

Dobijeni rezultati na eksperimentima sa CEC 2019 funkcijama su upoređeni sa drugim metaheurističkim tehnikama (EHOI, EHO, SCA, SSA, GOA, WOA, BBO, MFO, PSO). U komparativnoj analizi se upoređuju prosečne vrednosti ciljnih funkcija i standardna devijacija među različitim pristupima, pri čemu predložena metoda nadmašuje sve druge najsavremenije metode. Dodatno, korišćeni su neparametarski statistički testovi za dokaz značajne razlike između algoritama, konkretno korišćeno je Fridmanov test i Imanov Davenportov test. Nakon sprovođenja neparametarskih statističkih testova, Holmova *step-down* procedura je korišćena kao *post-hoc* procedura. Dobijeni statistički rezultati dokazuju značajnu razliku između rezultata dobijenih HEFFF metodom u odnosu na druge konkurentske pristupe.

U drugoj simulaciji, algoritam je primenjen za problem klasterovanja tekstualnih dokumenata, gde se šest standardnih tekstualnih skupova podataka (*CSTR*, *20 Newsgroups*, *Tr41*, *Tr12*, *Wap* i *Classic 4*) koristi za potrebe evaulacije performanse. Dobijeni rezultati su upoređeni s drugim metaheurističkim i nemetaheurističkim pristupima. Od metaheurističkih pristupa upoređene su sledeće metode: KHA, HS, PSO, GA, MVO, H-PSO, H-GA, H-MVO1, H-MVO2. Od nemetaheurističkih metoda upoređen je pristup <sup>13</sup> *K-means*, *K-means++*, *DBSCAN*, *Agglomerative* i *Spectral*. Za potrebe evaulacije u eksperimentu klasterovanja tekstualnih dokumenata su korišćene različite mere: stopa greške, tačnost, preciznost, odziv, F-mera, čistoća i entropija. Predloženi hibridni metod HEFFF pokazuje mnogo bolje rezultate u odnosu na druge najsavremenije tehnike i stoga možemo zaključiti da je pogodan za problem klasterovanja tekstualnih dokumenata. Razvoj nove poboljšane hibridne metode inteligencije roja sa dodatnom kombinacijom tradicionalnog algoritma K-srednjih vrednosti dovodi do okvira koji pruža najbolje performanse za klasterovanje dokumenata zasnovanih na tekstualnim podacima, što se vidi u dobijenim rezultatima. Predloženi metod efikasno kreira grupe u kojima su sadržaji tekstova vrlo slični, dok

je sličnost među različitim grupama niska.

Na osnovu korišćenih CEC 2019 test funkcija, predloženi metod daje odlične rezultate, nadmašuje rezultate drugih savremenih tehnika, stoga predloženi metod može biti pogodan i za primenu drugih NP-teških problema.

## 8 Literatura

- [1] Xu, D., Tian, Y.: A comprehensive survey of clustering algorithms. *Annals of Data Science* 2(2), 165–193 (2015)
- [2] Sun, W., Tang, M., Zhang, L., Huo, Z., Shu, L.: A survey of using swarm intelligence algorithms in iot. *Sensors* 20(5) (2020), <https://www.mdpi.com/1424-8220/20/5/1420>
- [3] Hrosik, R.C., Tuba, E., Dolicanin, E., Jovanovic, R., Tuba, M.: Brain image segmentation based on firefly algorithm combined with k-means clustering. *Studies in Informatics and Control*, 28(2), 167–176 (2019)
- [4] Tuba, E., Strumberger, I., Bezdan, T., Bacanin, N., Tuba, M.: Classification and feature selection method for medical datasets by brain storm optimization algorithm and support vector machine. *Procedia Computer Science* 162, 307–315 (2019)
- [5] Tuba, E., Strumberger, I., Bacanin, N., Bezdan, T., Tuba, M.: Image clustering by generative adversarial optimization and advanced clustering criteria. In: *International Conference on Swarm Intelligence*. pp. 465–475. Springer (2020)
- [6] Brajevic, I., Tuba, M., Bacanin, N.: Multilevel image thresholding selection based on the cuckoo search algorithm. In: *Proceedings of the 5th International Conference on Visualization, Imaging and Simulation (VIS'12)*, Sliema, Malta. pp. 217–222 (2012)
- [7] Lichtblau, D., Stoean, C.: Cancer diagnosis through a tandem of classifiers for digitized histopathological slides. *PLOS ONE* 14(1), 1–20 (01 2019), <https://doi.org/10.1371/journal.pone.0209274>
- [8] Bezdan, T., Cvetnic, D., Gajic, L., Zivkovic, M., Strumberger, I., Bacanin, N.: Feature selection by firefly algorithm with improved initialization strategy. In: *7th Conference on the Engineering of Computer Based Systems. ECBS 2021*, Association for Computing Machinery, New York, NY, USA (2021), <https://doi.org/10.1145/3459960.3459974>

- [9] Rana, S., Jasola, S., Kumar, R.: A review on particle swarm optimization algorithms and their applications to data clustering. *Artificial Intelligence Review* 35(3), 211–222 (2011)
- [10] Strumberger, I., Tuba, E., Bacanin, N., Zivkovic, M., Beko, M., Tuba, M.: Designing convolutional neural network architecture by the firefly algorithm. In: 2019 International Young Engineers Forum (YEF-ECE). pp. 59–65 (May 2019)
- [11] Bezdan, T., Tuba, E., Strumberger, I., Bacanin, N., Tuba, M.: Automatically designing convolutional neural network architecture with artificial flora algorithm. In: Tuba, M., Akashe, S., Joshi, A. (eds.) *ICT Systems and Sustainability*. pp. 371–378. Springer Singapore, Singapore (2020)
- [12] Bacanin, N., Bezdan, T., Tuba, E., Strumberger, I., Tuba, M.: Monarch Butterfly Optimization Based Convolutional Neural Network Design. *Mathematics* 8(6), 936 (2020), publisher: Multidisciplinary Digital Publishing Institute
- [13] Bacanin, N., Bezdan, T., Tuba, E., Strumberger, I., Tuba, M.: Optimizing convolutional neural network hyperparameters by enhanced swarm intelligence metaheuristics. *Algorithms* 13(3) (2020), <https://www.mdpi.com/1999-4893/13/3/67>
- [14] Stoean, R.: Analysis on the potential of an ea-surrogate modelling tandem for deep learning parametrization: an example for cancer classification from medical images. *Neural Computing and Applications* 32, 313–322 (2018)
- [15] Strumberger, I., Minovic, M., Tuba, M., Bacanin, N.: Performance of elephant herding optimization and tree growth algorithm adapted for node localization in wireless sensor networks. *Sensors* 19(11), 2515 (2019)
- [16] Zivkovic, M., Bacanin, N., Tuba, E., Strumberger, I., Bezdan, T., Tuba, M.: Wireless Sensor Networks Life Time Optimization Based on the Improved Firefly Algorithm. In: 2020 International Wireless Communications and Mobile Computing (IWCMC). pp. 1176–1181. IEEE (2020)

- [17] Bacanin, N., Bezdan, T., Tuba, E., Strumberger, I., Tuba, M., Zivkovic, M.: Task Scheduling in Cloud Computing Environment by Grey Wolf Optimizer. In: 2019 27th Telecommunications Forum TELFOR). pp. 1–4. IEEE (2019)
- [18] Bacanin, N., Tuba, E., Bezdan, T., Strumberger, I., Tuba, M.: Artificial flora optimization algorithm for task scheduling in cloud computing environment. In: International Conference on Intelligent Data Engineering and Automated Learning. pp. 437–445. Springer (2019)
- [19] Heidari, A.A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M., Chen, H.: Harris hawks optimization: Algorithm and applications. *Future Generation Computer Systems* 97, 849–872 (2019), publisher: Elsevier
- [20] Bezdan, T., Zivkovic, M., Antonijevic, M., Zivkovic, T., Bacanin, N.: Enhanced Flower Pollination Algorithm for Task Scheduling in Cloud Computing Environment. In: *Machine Learning for Predictive Analysis*, pp. 163–171. Springer (2020)
- [21] Bezdan, T., Zivkovic, M., Tuba, E., Strumberger, I., Bacanin, N., Tuba, M.: Multi-objective task scheduling in cloud computing environment by hybridized bat algorithm. In: Kahraman, C., Cevik Onar, S., Oztaysi, B., Sari, I.U., Cebi, S., Tolga, A.C. (eds.) *Intelligent and Fuzzy Techniques: Smart and Innovative Solutions*. pp. 718–725. Springer International Publishing, Cham (2021)
- [22] Strumberger, I., Bacanin, N., Tuba, M., Tuba, E.: Resource scheduling in cloud computing based on a hybridized whale optimization algorithm. *Applied Sciences* 9(22), 4893 (2019)
- [23] Bitam, S., Mellouk, A., Zeadally, S.: Bio-inspired routing algorithms survey for vehicular ad hoc networks. *IEEE Communications Surveys Tutorials* 17(2), 843–867 (Secondquarter 2015)
- [24] Marinakis, Y., Iordanidou, G.R., Marinaki, M.: Particle swarm optimization for the vehicle routing problem with stochastic demands. *Applied Soft Computing* 13(4), 1693 – 1704 (2013), <http://www.sciencedirect.com/science/article/pii/S1568494613000240>

- [25] Tuba, E., Strumberger, I., Zivkovic, D., Bacanin, N., Tuba, M.: Mobile robot path planning by improved brain storm optimization algorithm. In: 2018 IEEE Congress on Evolutionary Computation (CEC). pp. 1–8 (July 2018)
- [26] Lodi, A., Martello, S., Vigo, D.: Heuristic and metaheuristic approaches for a class of two-dimensional bin packing problems. *INFORMS Journal on Computing* 11(4), 345–357 (1999)
- [27] Madni, S.H.H., Latiff, M.S.A., Coulibaly, Y., Abdulhamid, S.M.: An appraisal of meta-heuristic resource allocation techniques for iaas cloud. *Indian Journal of Science and Technology* 9(4), 1–14 (2016)
- [28] Bezdan, T., Zivkovic, M., Tuba, E., Strumberger, I., Bacanin, N., Tuba, M.: Glioma Brain Tumor Grade Classification from MRI Using Convolutional Neural Networks Designed by Modified FA. In: International Conference on Intelligent and Fuzzy Systems. pp. 955–963. Springer (2020)
- [29] Bacanin, N., Tuba, E., Bezdan, T., Strumberger, I., Jovanovic, R., Tuba, M.: Dropout probability estimation in convolutional neural networks by the enhanced bat algorithm. In: 2020 International Joint Conference on Neural Networks (IJCNN). pp. 1–7 (2020)
- [30] Gajic, L., Cvetnic, D., Zivkovic, M., Bezdan, T., Bacanin, N., Milosevic, S.: Multi-layer Perceptron Training Using Hybridized Bat Algorithm. In: Smys, S., Tavares, J.M.R.S., Bestak, R., Shi, F. (eds.) *Computational Vision and Bio-Inspired Computing*. pp. 689–705. Springer Singapore, Singapore (2021)
- [31] Milosevic, S., Bezdan, T., Zivkovic, M., Bacanin, N., Strumberger, I., Tuba, M.: Feed-forward neural network training by hybrid bat algorithm. In: *Modelling and Development of Intelligent Systems: 7th International Conference, MDIS 2020, Sibiu, Romania, October 22–24, 2020, Revised Selected Papers 7*. pp. 52–66. Springer International Publishing (2021)
- [32] Bacanin, N., Bezdan, T., Venkatachalam, K., Al-Turjman, F.: Optimized convolutional neural network by firefly algorithm for magnetic resonance image classification of glioma brain tumor grade. *Journal of Real-Time Image Processing* (apr 2021), <https://doi.org/10.1007/s11554-021-01106-x>

- [33] Griffis, S.E., Bell, J.E., Closs, D.J.: Metaheuristics in logistics and supply chain management. *Journal of Business Logistics* 33(2), 90–106 (2012)
- [34] Pan, W.T.: A new evolutionary computation approach: fruit fly optimization algorithm. In: 2011 Conference of Digital Technology and Innovation Management. pp. 382–391 (2011)
- [35] Pan, W.T.: A new fruit fly optimization algorithm: taking the financial distress model as an example. *Knowledge-Based Systems* 26, 69–74 (2012)
- [36] Yang, X.S.: Firefly algorithms for multimodal optimization. In: Watanabe, O., Zeugmann, T. (eds.) *Stochastic Algorithms: Foundations and Applications*. pp. 169–178. Springer Berlin Heidelberg, Berlin, Heidelberg (2009)
- [37] MacQueen, J.B.: Some methods for classification and analysis of multivariate observations. In: Cam, L.M.L., Neyman, J. (eds.) *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*. vol. 1, pp. 281–297. University of California Press (1967)
- [38] Xiao, W., Yang, Y., Xing, H., Meng, X.: Clustering algorithm based on fruit fly optimization. In: *Rough Sets and Knowledge Technology: 10th International Conference, RSKT 2015, Held as Part of the International Joint Conference on Rough Sets, IJCRS 2015, Tianjin, China, November 20–23, 2015, Proceedings* 10. pp. 408–419. Springer (2015)
- [39] Bezdan, T., Stoean, C., Naamany, A.A., Bacanin, N., Rashid, T.A., Zivkovic, M., Venkatachalam, K.: Hybrid fruit-fly optimization algorithm with k-means for text document clustering. *Mathematics* 9(16), 1929 (2021)
- [40] Jamil, M., Yang, X.: A literature survey of benchmark functions for global optimization problems. *CoRR* abs/1308.4008 (2013), <http://arxiv.org/abs/1308.4008>
- [41] Glover, F.: Future paths for integer programming and links to artificial intelligence. *Comput. Oper. Res.* 13(5), 533–549 (may 1986), [http://dx.doi.org/10.1016/0305-0548\(86\)90048-1](http://dx.doi.org/10.1016/0305-0548(86)90048-1)

- [42] Fogel, D., Society, I.C.I.: Evolutionary computation: toward a new philosophy of machine intelligence. IEEE series on computational intelligence, Wiley (2006), <https://books.google.rs/books?id=BvdQAAAAAAAJ>
- [43] Beyer, H.G., Schwefel, H.P.: Evolution strategies – a comprehensive introduction. Natural Computing 1(1), 3–52 (Mar 2002), <https://doi.org/10.1023/A:1015059928466>
- [44] Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st edn. (1989)
- [45] Park, H.S., Jun, C.H.: A simple and fast algorithm for k-medoids clustering. Expert systems with applications 36(2), 3336–3341 (2009)
- [46] Kaufman, L., Rousseeuw, P.J.: Finding groups in data: an introduction to cluster analysis, vol. 344. John Wiley & Sons (2009)
- [47] Ng, R.T., Han, J.: Clarans: A method for clustering objects for spatial data mining. IEEE transactions on knowledge and data engineering 14(5), 1003–1016 (2002)
- [48] Zhang, T., Ramakrishnan, R., Livny, M.: Birch: an efficient data clustering method for very large databases. ACM sigmod record 25(2), 103–114 (1996)
- [49] Guha, S., Rastogi, R., Shim, K.: Cure: An efficient clustering algorithm for large databases. ACM Sigmod record 27(2), 73–84 (1998)
- [50] Guha, S., Rastogi, R., Shim, K.: Rock: A robust clustering algorithm for categorical attributes. Information systems 25(5), 345–366 (2000)
- [51] Karypis, G., Han, E.H., Kumar, V.: Chameleon: Hierarchical clustering using dynamic modeling. Computer 32(8), 68–75 (1999)
- [52] Bezdek, J.C., Ehrlich, R., Full, W.: Fcm: The fuzzy c-means clustering algorithm. Computers & geosciences 10(2-3), 191–203 (1984)
- [53] Dave, R.N., Bhaswan, K.: Adaptive fuzzy c-shells clustering and detection of ellipses. IEEE Transactions on Neural Networks 3(5), 643–662 (1992)

- [54] Yager, R.R., Filev, D.P.: Approximate clustering via the mountain method. *IEEE Transactions on systems, man, and Cybernetics* 24(8), 1279–1284 (1994)
- [55] Xu, X., Ester, M., Kriegel, H.P., Sander, J.: A distribution-based clustering algorithm for mining in large spatial databases. In: *Proceedings 14th International Conference on Data Engineering*. pp. 324–331. IEEE (1998)
- [56] Rasmussen, C.E., et al.: The infinite gaussian mixture model. In: *NIPS*. vol. 12, pp. 554–560 (1999)
- [57] Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*. p. 226–231. KDD’96, AAAI Press (1996)
- [58] Ankerst, M., Breunig, M.M., Kriegel, H.P., Sander, J.: Optics: Ordering points to identify the clustering structure. *ACM Sigmod record* 28(2), 49–60 (1999)
- [59] Comaniciu, D., Meer, P.: Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on pattern analysis and machine intelligence* 24(5), 603–619 (2002)
- [60] Sharan, R., Shamir, R.: Click: A clustering algorithm with applications to gene expression analysis. *Proceedings International Conference on Intelligent Systems for Molecular Biology ; ISMB. International Conference on Intelligent Systems for Molecular Biology* 8, 307–16 (02 2000)
- [61] Gupta, M.K., Chandra, P.: A comparative study of clustering algorithms. In: *2019 6th International Conference on Computing for Sustainable Global Development (INDIACom)*. pp. 801–805. IEEE (2019)
- [62] Wang, W., Yang, J., Muntz, R., et al.: Sting: A statistical information grid approach to spatial data mining. In: *VLDB*. vol. 97, pp. 186–195 (1997)
- [63] Agrawal, R., Gehrke, J., Gunopulos, D., Raghavan, P.: Automatic subspace clustering of high dimensional data for data mining applications. In: *Proceedings of the 1998 ACM SIGMOD international conference on Management of data*. pp. 94–105 (1998)

- [64] Barbará, D., Chen, P.: Using the fractal dimension to cluster datasets. In: Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 260–264 (2000)
- [65] Fisher, D.H.: Knowledge acquisition via incremental conceptual clustering. *Machine learning* 2(2), 139–172 (1987)
- [66] Hasanzadeh, E., Poyan, M., Rokny, H.A.: Text clustering on latent semantic indexing with particle swarm optimization (pso) algorithm. *International Journal of Physical Sciences* 7 (2012)
- [67] Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of ICNN'95-international conference on neural networks. vol. 4, pp. 1942–1948. IEEE (1995)
- [68] Karol, S., Mangat, V.: Evaluation of text document clustering approach based on particle swarm optimization. *Open Computer Science* 3(2), 69–90 (2013), <https://doi.org/10.2478/s13537-013-0104-2>
- [69] Sarkar, S., Roy, A., Purkayastha, B.S.: A comparative analysis of particle swarm optimization and k-means algorithm for text clustering using nepali wordnet (2014)
- [70] Aghdam, M.H., Heidari, S.: Feature selection using particle swarm optimization in text categorization. *Journal of Artificial Intelligence and Soft Computing Research* 5(4), 231–238 (2015), <https://doi.org/10.1515/jaiscr-2015-0031>
- [71] Song, W., Qiao, Y., Park, S.C., Qian, X.: A hybrid evolutionary computation approach with its application for optimizing text document clustering. *Expert Systems with Applications* 42(5), 2517–2524 (2015), <https://www.sciencedirect.com/science/article/pii/S0957417414006861>
- [72] Holland, J.H.: Genetic algorithms. *Scientific american* 267(1), 66–73 (1992)
- [73] Bharti, K.K., Singh, P.K.: Opposition chaotic fitness mutation based adaptive inertia weight bpsa for feature selection in text clustering. *Applied Soft Computing* 43, 20–34 (2016), <https://www.sciencedirect.com/science/article/pii/S1568494616300060>

- [74] Abualigah, L.M., Khader, A.T.A., Al-Betar, M.A., Hanandeh, E.S.: Unsupervised text feature selection technique based on particle swarm optimization algorithm for improving the text clustering (2017)
- [75] Daoud, A.S., Sallam, A., Wheed, M.E.: Improving arabic document clustering using k-means algorithm and particle swarm optimization. In: 2017 Intelligent Systems Conference (IntelliSys). pp. 879–885 (2017)
- [76] Kushwaha, N., Pant, M.: Link based bpsso for feature selection in big data text clustering. *Future Generation Computer Systems* 82, 190–199 (2018), <https://www.sciencedirect.com/science/article/pii/S0167739X17321854>
- [77] Janani, R., Vijayarani, S.: Text document clustering using spectral clustering algorithm with particle swarm optimization. *Expert Systems with Applications* 134, 192–200 (2019), <https://www.sciencedirect.com/science/article/pii/S0957417419303598>
- [78] Song, W., Park, S.C.: Genetic algorithm for text clustering based on latent semantic indexing. *Computers and Mathematics with Applications* 57(11), 1901–1907 (2009), <https://www.sciencedirect.com/science/article/pii/S0898122108005300>, proceedings of the International Conference
- [79] Premalatha, K., Natarajan, A., et al.: Genetic algorithm for document clustering with simultaneous and ranked mutation. *Modern Applied Science* 3(2), 75 (2009)
- [80] Song, W., Li, C.H., Park, S.C.: Genetic algorithm for text clustering using ontology and evaluating the validity of various semantic similarity measures. *Expert Systems with Applications* 36(5), 9095–9104 (2009), <https://www.sciencedirect.com/science/article/pii/S0957417408009123>
- [81] Lee, J.S., Choi, L.C., Park, S.C.: Document clustering using multi-objective genetic algorithm with different feature selection methods. In: *International Workshop on Semantic Interoperability*. vol. 2, pp. 101–110. SCITEPRESS (2011)

- [82] Chun-hong, W., Li-Li, N., Yao-Peng, R.: Research on the text clustering algorithm based on latent semantic analysis and optimization. In: 2011 IEEE International Conference on Computer Science and Automation Engineering. vol. 4, pp. 470–473. IEEE (2011)
- [83] Shi, K., Li, L.: High performance genetic algorithm based text clustering using parts of speech and outlier elimination. *Applied Intelligence* 38, 511–519 (2013)
- [84] Casillas, A., de Lena, M.T.G., Martínez, R.: Sampling and feature selection in a genetic algorithm for document clustering. In: Gelbukh, A. (ed.) *Computational Linguistics and Intelligent Text Processing*. pp. 601–612. Springer Berlin Heidelberg, Berlin, Heidelberg (2004)
- [85] Divyashree, G., Rayar, G.: Comparison between k-means and genetic algorithm in text document clustering
- [86] Karaa, W.B.A., Ashour, A.S., Sassi, D.B., Roy, P., Kausar, N., Dey, N.: MEDLINE Text Mining: An Enhancement Genetic Algorithm Based Approach for Document Clustering, pp. 267–287. Springer International Publishing, Cham (2016), [https://doi.org/10.1007/978-3-319-21212-8\\_12](https://doi.org/10.1007/978-3-319-21212-8_12)
- [87] Garg, N., Gupta, R.K.: Performance evaluation of new text mining method based on ga and k-means clustering algorithm. In: Choudhary, R.K., Mandal, J.K., Bhattacharyya, D. (eds.) *Advanced Computing and Communication Technologies*. pp. 23–30. Springer Singapore, Singapore (2018)
- [88] Mustafi, D., Mustafi, A., Sahoo, G.: A novel approach to text clustering using genetic algorithm based on the nearest neighbour heuristic. *International Journal of Computers and Applications* 44(3), 291–303 (2022)
- [89] Al-Jadir, I., Wong, K.W., Fung, C.C., Xie, H.: Adaptive crossover memetic differential harmony search for optimizing document clustering. In: Cheng, L., Leung, A.C.S., Ozawa, S. (eds.) *Neural Information Processing*. pp. 509–518. Springer International Publishing, Cham (2018)
- [90] Das, S., Suganthan, P.N.: Differential evolution: A survey of the state-of-the-art. *IEEE transactions on evolutionary computation* 15(1), 4–31 (2010)

- [91] Mustafa, H.M., Ayob, M., Albashish, D., Abu-Taleb, S.: Solving text clustering problem using a memetic differential evolution algorithm. *PLoS One* 15(6), e0232816 (2020)
- [92] Moscato, P., et al.: On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms. *Caltech concurrent computation program, C3P Report 826*(1989), 37 (1989)
- [93] Saini, N., Bansal, D., Saha, S., Bhattacharyya, P.: Multi-objective multi-view based search result clustering using differential evolution framework. *Expert Systems with Applications* 168, 114299 (2021), <https://www.sciencedirect.com/science/article/pii/S0957417420309994>
- [94] Mirjalili, S., Mirjalili, S.M., Lewis, A.: Grey wolf optimizer. *Advances in Engineering Software* 69, 46–61 (2014), <https://www.sciencedirect.com/science/article/pii/S0965997813001853>
- [95] Rashaideh, H., Sawaie, A., Al-Betar, M.A., Abualigah, L.M., Al-laham, M.M., Al-Khatib, R.M., Braik, M.: A grey wolf optimizer for text document clustering. *Journal of Intelligent Systems* 29(1), 814–830 (2020), <https://doi.org/10.1515/jisys-2018-0194>
- [96] Vidyadhari, C., Sandhya, N., Premchand, P.: A semantic word processing using enhanced cat swarm optimization algorithm for automatic text clustering. *Multimedia Research* 2(4), 23–32 (2019)
- [97] Abasi, A.K., Khader, A.T., Al-Betar, M.A., Naim, S., Makhadmeh, S.N., Alyasseri, Z.A.A.: An improved text feature selection for clustering using binary grey wolf optimizer. In: *Proceedings of the 11th National Technical Seminar on Unmanned System Technology 2019: NUSYS'19*. pp. 503–516. Springer (2021)
- [98] Yang, X.S., Deb, S.: Cuckoo search via lévy flights. In: *2009 World congress on nature & biologically inspired computing (NaBIC)*. pp. 210–214. Ieee (2009)

- [99] Zaw, M.M., Mon, E.E.: Web Document Clustering by Using PSO-Based Cuckoo Search Clustering Algorithm, pp. 263–281. Springer International Publishing, Cham (2015), [https://doi.org/10.1007/978-3-319-13826-8\\_14](https://doi.org/10.1007/978-3-319-13826-8_14)
- [100] Boushaki, S.I., Kamel, N., Bendjeghaba, O.: Improved cuckoo search algorithm for document clustering. In: Amine, A., Bellatreche, L., Elberrichi, Z., Neuhold, E.J., Wrembel, R. (eds.) Computer Science and Its Applications. pp. 217–228. Springer International Publishing, Cham (2015)
- [101] Ishak Boushaki, S., Kamel, N., Bendjeghaba, O.: High-dimensional text datasets clustering algorithm based on cuckoo search and latent semantic indexing. *Journal of Information & Knowledge Management* 17(03), 1850033 (2018)
- [102] Boushaki, S.I., Bendjeghaba, O., Brakta, N.: Document clustering analysis based on hybrid cuckoo search and k-means algorithm. In: 2021 IEEE 12th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON). pp. 0058–0062 (2021)
- [103] Mohammed, A.J., Yusof, Y., Husni, H.: Gf-clust: A nature-inspired algorithm for automatic text clustering. *Journal of Information and Communication Technology (JICT)* 15(1), 57–81 (2016)
- [104] Mohammed, A.J., Yusof, Y., Husni, H.: Integrated bisect k-means and firefly algorithm for hierarchical text clustering. *Journal of Engineering and Applied Sciences* 11(3), 522–527 (2016)
- [105] Abualigah, L.M., Khader, A.T., Al-Betar, M.A., Awadallah, M.A.: A krill herd algorithm for efficient text documents clustering. In: 2016 IEEE Symposium on Computer Applications and Industrial Electronics (ISCAIE). pp. 67–72 (2016)
- [106] Gandomi, A.H., Alavi, A.H.: Krill herd: a new bio-inspired optimization algorithm. *Communications in nonlinear science and numerical simulation* 17(12), 4831–4845 (2012)
- [107] Geem, Z.W., Kim, J.H., Loganathan, G.V.: A new heuristic optimization algorithm: harmony search. *simulation* 76(2), 60–68 (2001)

- [108] Abualigah, L.M., Khader, A.T., Hanandeh, E.S.: Hybrid clustering analysis using improved krill herd algorithm. *Applied Intelligence* 48, 4047–4071 (2018)
- [109] Abualigah, L.M., Khader, A.T., Al-Betar, M.A.: Multi-objectives-based text clustering technique using k-mean algorithm. In: 2016 7th International Conference on Computer Science and Information Technology (CSIT). pp. 1–6. IEEE (2016)
- [110] Abualigah, L.M.Q., et al.: Feature selection and enhanced krill herd algorithm for text document clustering (2019)
- [111] Chandran, T.R., Reddy, A., Janet, B.: A social spider optimization approach for clustering text documents. In: 2016 2nd International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB). pp. 22–26. IEEE (2016)
- [112] Cuevas, E., Cienfuegos, M., Zaldívar, D., Pérez-Cisneros, M.: A swarm optimization algorithm inspired in the behavior of the social-spider. *Expert Systems with Applications* 40(16), 6374–6384 (2013)
- [113] Chandran, T.R., Reddy, A., Janet, B.: Text clustering quality improvement using a hybrid social spider optimization. *International Journal of Applied Engineering Research* 12(6), 995–1008 (2017)
- [114] Gopal, J., Brunda, S.S.: Text clustering algorithm using fuzzy whale optimization algorithm. *International Journal of Intelligent Engineering and Systems* (2019)
- [115] Mirjalili, S., Lewis, A.: The whale optimization algorithm. *Advances in engineering software* 95, 51–67 (2016)
- [116] Sawarn, S., Deepak, G.: An approach for document clustering using semantic similarity and whale optimization. In: Musleh Al-Sartawi, A.M., Razzaque, A., Kamal, M.M. (eds.) *Artificial Intelligence Systems and the Internet of Things in the Digital Era*. pp. 322–333. Springer International Publishing, Cham (2021)
- [117] Cobo, A., Rocha, R.: Document management with ant colony optimization metaheuristic: A fuzzy text clustering approach using pheromone trails. In:

- Gaspar-Cunha, A., Takahashi, R., Schaefer, G., Costa, L. (eds.) *Soft Computing in Industrial Applications*. pp. 261–270. Springer Berlin Heidelberg, Berlin, Heidelberg (2011)
- [118] Dorigo, M., Birattari, M., Stutzle, T.: Ant colony optimization. *IEEE computational intelligence magazine* 1(4), 28–39 (2006)
- [119] Azaryuon, K., Fakhar, B.: A novel document clustering algorithm based on ant colony optimization algorithm. *Journal of mathematics and computer Science* 7, 171–180 (2013)
- [120] Nema, P., Sharma, V.: Multi-label text categorization based on feature optimization using ant colony optimization and relevance clustering technique. In: *2015 International Conference on Computers, Communications, and Systems (ICCCS)*. pp. 1–5. IEEE (2015)
- [121] Cobos, C., Andrade, J., Constain, W., Mendoza, M., León, E.: Web document clustering based on global-best harmony search, k-means, frequent term sets and bayesian information criterion. In: *IEEE congress on evolutionary computation*. pp. 1–8. IEEE (2010)
- [122] Forsati, R., Mahdavi, M., Shamsfard, M., Reza Meybodi, M.: Efficient stochastic algorithms for document clustering. *Information Sciences* 220, 269–291 (2013), <https://www.sciencedirect.com/science/article/pii/S0020025512004975>, online Fuzzy Machine Learning and Data Mining
- [123] Devi, M.P.A., Latha, T., Sulochana, C.H.: Iterative thresholding based image segmentation using 2d improved otsu algorithm. In: *2015 Global Conference on Communication Technologies (GCCT)*. pp. 145–149 (2015)
- [124] Devi, S.S., Shanmugam, A.: Hybridized harmony search method for text clustering using concept factorization. *Int. J. Adv. Comput. Technol* pp. 320–327 (2016)
- [125] Mirjalili, S., Mirjalili, S.M., Hatamlou, A.: Multi-verse optimizer: a nature-inspired algorithm for global optimization. *Neural Computing and Applications* 27, 495–513 (2016)

- [126] Abasi, A.K., Khader, A.T.A., Al-Betar, M.A., Naim, S., Makhadmeh, S.N., Alyasseri, Z.A.A.: Link-based multi-verse optimizer for text documents clustering. *Appl. Soft Comput.* 87, 106002 (2020)
- [127] Abasi, A., Khader, A., Al-Betar, M., Naim, S., Awadallah, M., Alomari, O.: Text documents clustering using modified multi-verse optimizer. *International Journal of Electrical and Computer Engineering* 10(6) (2020)
- [128] Abasi, A.K., Khader, A.T., Al-Betar, M.A., Naim, S., Alyasseri, Z.A.A., Makhadmeh, S.N.: A novel hybrid multi-verse optimizer with k-means for text documents clustering. *Neural Comput. Appl.* 32(23), 17703–17729 (2020)
- [129] Mirhosseini, M.: A clustering approach using a combination of gravitational search algorithm and k-harmonic means and its application in text document clustering. *Turkish Journal of Electrical Engineering and Computer Sciences* 25(2), 1251–1262 (2017)
- [130] Rashedi, E., Nezamabadi-Pour, H., Saryazdi, S.: Gsa: a gravitational search algorithm. *Information sciences* 179(13), 2232–2248 (2009)
- [131] Chu, S.C., Tsai, P.W., Pan, J.S.: Cat swarm optimization. In: *PRICAI 2006: Trends in Artificial Intelligence: 9th Pacific Rim International Conference on Artificial Intelligence Guilin, China, August 7-11, 2006 Proceedings* 9. pp. 854–858. Springer (2006)
- [132] Thirumoorthy, K., Muneeswaran, K.: A hybrid approach for text document clustering using jaya optimization algorithm. *Expert Systems with Applications* 178, 115040 (2021), <https://www.sciencedirect.com/science/article/pii/S0957417421004814>
- [133] Rao, R.: Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems. *International Journal of Industrial Engineering Computations* 7(1), 19–34 (2016)
- [134] Abasi, A.K., Khader, A.T., Al-Betar, M.A., Alyasseri, Z.A.A., Makhadmeh, S.N., Al-laham, M., Naim, S.: A Hybrid Salp Swarm Algorithm with Hill Clim-

- p. 129–161. Springer Singapore, Singapore (2021),
- [https://doi.org/10.1007/978-981-33-4191-3\\_6](https://doi.org/10.1007/978-981-33-4191-3_6)
- [135] Mirjalili, S., Gandomi, A.H., Mirjalili, S.Z., Saremi, S., Faris, H., Mirjalili, S.M.: Salp swarm algorithm: A bio-inspired optimizer for engineering design problems. *Advances in engineering software* 114, 163–191 (2017)
  - [136] Perumal, P., Mathivanan, B.: Type2 ifc with soa for topic detection and document clustering analysis (2021)
  - [137] Dhiman, G., Kumar, V.: Seagull optimization algorithm: Theory and its applications for large-scale industrial engineering problems. *Knowledge-based systems* 165, 169–196 (2019)
  - [138] Strumberger, I., Tuba, E., Bacanin, N., Zivkovic, M., Beko, M., Tuba, M.: Designing convolutional neural network architecture by the firefly algorithm. In: *Proceedings of the 2019 International Young Engineers Forum (YEF-ECE)*, Costa da Caparica, Portugal. pp. 59–65 (2019)
  - [139] Bezdan, T., Zivkovic, M., Tuba, E., Strumberger, I., Bacanin, N., Tuba, M.: Glioma brain tumor grade classification from mri using convolutional neural networks designed by modified fa. In: *International conference on intelligent and fuzzy systems*. pp. 955–963. Springer (2020)
  - [140] Bacanin, N., Bezdan, T., Venkatachalam, K., Al-Turjman, F.: Optimized convolutional neural network by firefly algorithm for magnetic resonance image classification of glioma brain tumor grade. *Journal of Real-Time Image Processing* 18(4), 1085–1098 (2021)
  - [141] Tuba, M., Bacanin, N.: Jpeg quantization tables selection by the firefly algorithm. In: *2014 International Conference on Multimedia Computing and Systems (ICMCS)*. pp. 153–158. IEEE (2014)
  - [142] Bacanin, N., Tuba, M.: Firefly algorithm for cardinality constrained mean-variance portfolio optimization problem with entropy diversity constraint. *The Scientific World Journal* 2014 (2014)

- [143] Bacanin, N., Tuba, M., Jovanovic, R.: Hierarchical multiobjective rfid network planning using firefly algorithm. In: 2015 International Conference on Information and Communication Technology Research (ICTRC). pp. 282–285 (May 2015)
- [144] Strumberger, I., Bacanin, N., Tuba, M.: Enhanced firefly algorithm for constrained numerical optimization. In: 2017 IEEE Congress on Evolutionary Computation (CEC). pp. 2120–2127. Ieee (2017)
- [145] Bezdan, T., Cvetnic, D., Gajic, L., Zivkovic, M., Strumberger, I., Bacanin, N.: Feature selection by firefly algorithm with improved initialization strategy. In: 7th conference on the engineering of computer based systems. pp. 1–8 (2021)
- [146] Bacanin, N., Stoean, R., Zivkovic, M., Petrovic, A., Rashid, T.A., Bezdan, T.: Performance of a novel chaotic firefly algorithm with enhanced exploration for tackling global optimization problems: Application for dropout regularization. *Mathematics* 9(21), 2705 (2021)
- [147] Zivkovic, M., Bacanin, N., Tuba, E., Strumberger, I., Bezdan, T., Tuba, M.: Wireless sensor networks life time optimization based on the improved firefly algorithm. In: 2020 International Wireless Communications and Mobile Computing (IWCMC). pp. 1176–1181. IEEE (2020)
- [148] Tizhoosh, H.R.: Opposition-based learning: a new scheme for machine intelligence. In: International conference on computational intelligence for modelling, control and automation and international conference on intelligent agents, web technologies and internet commerce (CIMCA-IAWTIC'06). vol. 1, pp. 695–701. IEEE (2005)
- [149] Bezdan, T., Petrovic, A., Zivkovic, M., Strumberger, I., Devi, V.K., Bacanin, N.: Current best opposition-based learning salp swarm algorithm for global numerical optimization. In: 2021 zooming innovation in consumer technologies conference (ZINC). pp. 5–10. IEEE (2021)
- [150] Bacanin, N., Arnaut, U., Zivkovic, M., Bezdan, T., Rashid, T.A.: Energy efficient clustering in wireless sensor networks by opposition-based initialization bat

- algorithm. In: Computer Networks and Inventive Communication Technologies: Proceedings of Fourth ICCNCT 2021, pp. 1–16. Springer (2021)
- [151] Tizhoosh, H.R., Ventresca, M.: Oppositional concepts in computational intelligence. In: Oppositional Concepts in Computational Intelligence (2008)
  - [152] Wang, H., Zhou, X., Sun, H., Yu, X., Zhao, J., Zhang, H., Cui, L.: Firefly algorithm with adaptive control parameters. *Soft Computing* 3, 5091—5102 (2017)
  - [153] Price, K., Awad, N., Ali, M., Suganthan, P.: Problem definitions and evaluation criteria for the 100-digit challenge special session and competition on single objective numerical optimization. In: Technical Report. Nanyang Technological University (2018)
  - [154] Muthusamy, H., Ravindran, S., Yaacob, S., Polat, K.: An improved elephant herding optimization using sine–cosine mechanism and opposition based learning for global optimization problems. *Expert Systems with Applications* 172, 114607 (2021)
  - [155] Wang, G.G., Deb, S., Coelho, L.d.S.: Elephant herding optimization. In: 2015 3rd international symposium on computational and business intelligence (ISC-BI). pp. 1–5. IEEE (2015)
  - [156] Mirjalili, S.: Sca: a sine cosine algorithm for solving optimization problems. *Knowledge-based systems* 96, 120–133 (2016)
  - [157] Saremi, S., Mirjalili, S., Lewis, A.: Grasshopper optimisation algorithm: theory and application. *Advances in engineering software* 105, 30–47 (2017)
  - [158] Mirjalili, S., Lewis, A.: The whale optimization algorithm. *Advances in engineering software* 95, 51–67 (2016)
  - [159] Simon, D.: Biogeography-based optimization. *IEEE transactions on evolutionary computation* 12(6), 702–713 (2008)
  - [160] Mirjalili, S.: Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. *Knowledge-based systems* 89, 228–249 (2015)

- [161] Friedman, M.: The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the american statistical association* 32(200), 675–701 (1937)
- [162] Friedman, M.: A comparison of alternative tests of significance for the problem of m rankings. *The Annals of Mathematical Statistics* 11(1), 86–92 (1940)
- [163] Iman, R.L., Davenport, J.M.: Approximations of the critical region of the fbietkan statistic. *Communications in Statistics-Theory and Methods* 9(6), 571–595 (1980)
- [164] Sheskin, D.J.: *Handbook of parametric and nonparametric statistical procedures*. Chapman and Hall/CRC (2020)
- [165] Voorhees, E.M., Buckland, L.P. (eds.): *Proceedings of The Seventeenth Text REtrieval Conference, TREC 2008*, Gaithersburg, Maryland, USA, November 18-21, 2008, NIST Special Publication, vol. 500-277. National Institute of Standards and Technology (NIST) (2008), [http://trec.nist.gov/pubs/trec17/t17\\_proceedings.html](http://trec.nist.gov/pubs/trec17/t17_proceedings.html)
- [166] Voorhees, E.M., Harman, D.K. (eds.): *Proceedings of The Fifth Text REtrieval Conference, TREC 1996*, Gaithersburg, Maryland, USA, November 20-22, 1996, NIST Special Publication, vol. 500-238. National Institute of Standards and Technology (NIST) (1996), [http://trec.nist.gov/pubs/trec5/t5\\_proceedings.html](http://trec.nist.gov/pubs/trec5/t5_proceedings.html)
- [167] Boley, D., Gini, M., Gross, R., Han, E.H., Hastings, K., Karypis, G., Kumar, V., Mobasher, B., Moore, J.: Document categorization and query generation on the world wide web using webace. *Artificial Intelligence Review* 13, 365–391 (1999)

# KLASTEROVANJE TEKSTUALNIH DOKUMENATA POBOLJŠANOM HIBRIDNOM METODOM METAHEURISTIKE I ALGORITMOM NENADGLEDANOG MAŠINSKOG UČENJA

ORIGINALITY REPORT

9%

SIMILARITY INDEX

## PRIMARY SOURCES

|   |                                                                                                                                                                                                                              |                  |
|---|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|
| 1 | <a href="https://mdpi-res.com">mdpi-res.com</a><br>Internet                                                                                                                                                                  | 352 words — 1%   |
| 2 | Ammar Kamal Abasi, Ahamad Tajudin Khader, Mohammed Azmi Al-Betar, Syibrah Naim et al. "A novel hybrid multi-verse optimizer with K-means for text documents clustering", Neural Computing and Applications, 2020<br>Crossref | 279 words — 1%   |
| 3 | <a href="https://pdfslide.net">pdfslide.net</a><br>Internet                                                                                                                                                                  | 128 words — 1%   |
| 4 | <a href="http://www.yuinfo.org">www.yuinfo.org</a><br>Internet                                                                                                                                                               | 128 words — 1%   |
| 5 | <a href="https://nardus.mpn.gov.rs">nardus.mpn.gov.rs</a><br>Internet                                                                                                                                                        | 118 words — < 1% |
| 6 | <a href="http://www.etrان.rs">www.etrان.rs</a><br>Internet                                                                                                                                                                   | 80 words — < 1%  |
| 7 | <a href="http://www.mdpi.com">www.mdpi.com</a><br>Internet                                                                                                                                                                   | 61 words — < 1%  |

8 Timea Bezdan, Catalin Stoean, Ahmed Al Naamany, Nebojsa Bacanin, Tarik A. Rashid, Miodrag Zivkovic, K. Venkatachalam. "Hybrid Fruit-Fly Optimization Algorithm with K-Means for Text Document Clustering", Mathematics, 2021

53 words — < 1%

Crossref

9 Ibrahim Hayatu Hassan, Abdullahi Mohammed, Yusuf Sahabi Ali, Isuwa Jeremiah, Silifat Adaramaja Abdulraheem. "Metaheuristic algorithms in text clustering", Elsevier BV, 2023

35 words — < 1%

Crossref

10 [hdl.handle.net](https://hdl.handle.net)

Internet

32 words — < 1%

11 Mohammed Azmi Al-Betar, Ammar Kamal Abasi, Ghazi Al-Naymat, Kamran Arshad, Sharif Naser Makhadmeh. "Bare-Bones Based Salp Swarm Algorithm for Text Document Clustering", IEEE Access, 2023

30 words — < 1%

Crossref

12 [singipedia.singidunum.ac.rs](https://singipedia.singidunum.ac.rs)

Internet

27 words — < 1%

13 Ammar Kamal Abasi, Ahamad Tajudin Khader, Mohammed Azmi Al-Betar, Syibrah Naim et al. "An ensemble topic extraction approach based on optimization clusters using hybrid multi-verse optimizer for scientific publications", Journal of Ambient Intelligence and Humanized Computing, 2020

26 words — < 1%

Crossref

14 [lume.ufrgs.br](https://lume.ufrgs.br)

Internet

26 words — < 1%

15 [res.mdpi.com](https://res.mdpi.com)

Internet

26 words — < 1%

16 [link.springer.com](https://link.springer.com)

Internet

25 words — < 1%

17 [ipfs.io](https://ipfs.io)

Internet

21 words — < 1%

18 [vbs.rs](https://vbs.rs)

Internet

21 words — < 1%

19 "HAAR - HALTEN", Walter de Gruyter GmbH, 1997

Crossref

20 words — < 1%

20 [www.tintakiado.hu](http://www.tintakiado.hu)

Internet

20 words — < 1%

21 [9lib.net](https://9lib.net)

Internet

18 words — < 1%

22 Senthil Kumaran Selvaraj, Jayakumar Kaliappan, Natarajan Muthuswamy, S. Ramesh Kumar, Muralimohan Cheepu. "Prognostics, Health Assessment, and Modelling of Material Removal Rate by EDM for Al 6061 and AISI 304 via Cockroach Swarm and Fruit Fly Optimization Approaches", Advances in Materials Science and Engineering, 2022

Crossref

17 words — < 1%

23 [bib.irb.hr](https://bib.irb.hr)

Internet

17 words — < 1%

24 [www.hindawi.com](http://www.hindawi.com)

Internet

17 words — < 1%

25 [repositorio.unb.br](https://repositorio.unb.br)

16 words — &lt; 1%

26 [repozitorij.etfos.hr](#)

Internet

16 words — &lt; 1%

27 [ebin.pub](#)

Internet

15 words — &lt; 1%

28 [repozitorij.fsb.unizg.hr](#)

Internet

15 words — &lt; 1%

29 Laith Mohammad Abualigah, Ahamad Tajudin Khader, Mohammed Azmi Al-Betar, Mohammed A. Awadallah. "A krill herd algorithm for efficient text documents clustering", 2016 IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE), 2016

Crossref

14 words — &lt; 1%

30 Yuzhen Li, Shihao Wang. "Differential evolution algorithm with elite archive and mutation strategies collaboration", Artificial Intelligence Review, 2019

Crossref

14 words — &lt; 1%

31 "PRICAI 2016: Trends in Artificial Intelligence", Springer Science and Business Media LLC, 2016

Crossref

13 words — &lt; 1%

32 [nure.ua](#)

Internet

13 words — &lt; 1%

33 "Advances in Artificial Intelligence – IBERAMIA 2012", Springer Science and Business Media LLC, 2012

Crossref

12 words — &lt; 1%

34 "Advances in Swarm Intelligence", Springer Science and Business Media LLC, 2017

12 words — &lt; 1%

35 Pawan Kumar Mandal. "A review of classical methods and Nature-Inspired Algorithms (NIAs) for optimization problems", Results in Control and Optimization, 2023

Crossref

12 words — &lt; 1%

36 [kongre.akademikiletisim.com](http://kongre.akademikiletisim.com)

Internet

12 words — &lt; 1%

37 Abubaker Younis, Abdalaziz Bakhit, Mahmoud Onsa, Mohsin Hashim. "A comprehensive and critical review of bio-inspired metaheuristic frameworks for extracting parameters of solar cell single and double diode models", Energy Reports, 2022

Crossref

11 words — &lt; 1%

38 Khan, R.A.. "Existence and approximation of solutions of second-order nonlinear four point boundary value problems", Nonlinear Analysis, 20051231

Crossref

11 words — &lt; 1%

39 [atorwithme.blogspot.com](http://atorwithme.blogspot.com)

Internet

11 words — &lt; 1%

40 [vdocuments.mx](http://vdocuments.mx)

Internet

11 words — &lt; 1%

41 Hathiram Nenavath, Dr Ravi Kumar Jatoth, Dr Swagatam Das. "A synergy of the sine-cosine algorithm and particle swarm optimizer for improved global optimization and object tracking", Swarm and Evolutionary Computation, 2018

Crossref

10 words — &lt; 1%

42 [docksci.com](http://docksci.com)

Internet

10 words — &lt; 1%

|    |                                                                                                                                                                                           |                 |
|----|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|
| 43 | <a href="http://edoc.site">edoc.site</a><br>Internet                                                                                                                                      | 10 words — < 1% |
| 44 | <a href="http://ms.copernicus.org">ms.copernicus.org</a><br>Internet                                                                                                                      | 10 words — < 1% |
| 45 | <a href="http://repositorio.unicamp.br">repositorio.unicamp.br</a><br>Internet                                                                                                            | 10 words — < 1% |
| 46 | <a href="http://www.viva-technology.org">www.viva-technology.org</a><br>Internet                                                                                                          | 10 words — < 1% |
| 47 | Dorothy Rajendran, Thankappan Sasilatha, Susai Rajendran, Senthil Selvaraj et al. "Application of machine learning in corrosion inhibition study", Zastita materijala, 2022<br>Crossref   | 9 words — < 1%  |
| 48 | Lecture Notes in Computer Science, 2011.<br>Crossref                                                                                                                                      | 9 words — < 1%  |
| 49 | Megha Varshney, Pravesh Kumar, Musrrat Ali, Yonis Gulzar. "Using the Grey Wolf Aquila Synergistic Algorithm for Design Problems in Structural Engineering", Biomimetics, 2024<br>Crossref | 9 words — < 1%  |
| 50 | <a href="http://aip.scitation.org">aip.scitation.org</a><br>Internet                                                                                                                      | 9 words — < 1%  |
| 51 | <a href="http://orca.cardiff.ac.uk">orca.cardiff.ac.uk</a><br>Internet                                                                                                                    | 9 words — < 1%  |
| 52 | <a href="http://ri4es.etf.bg.ac.rs">ri4es.etf.bg.ac.rs</a><br>Internet                                                                                                                    | 9 words — < 1%  |

- 
- 53 [www.scribd.com](https://www.scribd.com) 9 words — < 1%  
Internet
- 
- 54 Akash Saxena, Rajesh Kumar, Swagatam Das. " - 8 words — < 1%  
Chaotic map enabled Grey Wolf Optimizer ",  
Applied Soft Computing, 2018  
Crossref
- 
- 55 Bibi Aamirah Shafaa Emambocus, Muhammed 8 words — < 1%  
Basheer Jasser, Angela Amphawan, Ali Wagdy  
Mohamed. "An Optimized Discrete Dragonfly Algorithm  
Tackling the Low Exploitation Problem for Solving TSP",  
Mathematics, 2022  
Crossref
- 
- 56 Dalia Yousri, Seyedali Mirjalili, J.A. Tenreiro 8 words — < 1%  
Machado, Sudhakar Babu Thanikanti, Osama  
elbaksawi, Ahmed Fathy. "Efficient fractional-order modified  
Harris hawks optimizer for proton exchange membrane fuel  
cell modeling", Engineering Applications of Artificial  
Intelligence, 2021  
Crossref
- 
- 57 Gugulothu Venkanna, Dr K F Bharati. "Optimal Text 8 words — < 1%  
Document Clustering Enabled by Weighed  
Similarity Oriented Jaya With Grey Wolf Optimization  
Algorithm", The Computer Journal, 2021  
Crossref
- 
- 58 Qingyong Wang, Yun Zhou, Zehong Cao, Weiming 8 words — < 1%  
Zhang. "M2SPL: Generative multiview features  
with adaptive meta-self-paced sampling for class-imbalance  
learning", Expert Systems with Applications, 2022  
Crossref
- 
- 59 Velimir Stavljanin, Vesna Milicevic, Jasmina 8 words — < 1%  
Makuljevic. "Strategic approach to film marketing

- 
- 60 [c.coek.info](http://c.coek.info) 8 words — < 1%  
Internet
- 
- 61 [ppgcc.propesp.ufpa.br](http://ppgcc.propesp.ufpa.br) 8 words — < 1%  
Internet
- 
- 62 [www.cris.uns.ac.rs](http://www.cris.uns.ac.rs) 8 words — < 1%  
Internet
- 
- 63 [www.cs.uoi.gr](http://www.cs.uoi.gr) 8 words — < 1%  
Internet
- 
- 64 [www.ftn.uns.ac.rs](http://www.ftn.uns.ac.rs) 8 words — < 1%  
Internet
- 
- 65 [www.lejla-bm.com.ba](http://www.lejla-bm.com.ba) 8 words — < 1%  
Internet
- 
- 66 [www.proz.com](http://www.proz.com) 8 words — < 1%  
Internet
- 
- 67 Leilei Yu, Sian-Jheng Lin, Hanxu Hou, Zhengrui Li. "Reed-Solomon Coding Algorithms Based on Reed-Muller Transform for Any Number of Parities", IEEE Transactions on Computers, 2023 7 words — < 1%  
Crossref
- 
- 68 Nebojsa Bacanin, Miodrag Zivkovic, Fadi Al-Turjman, K. Venkatachalam, Pavel Trojovský, Ivana Strumberger, Timea Bezdan. "Hybridized sine cosine algorithm with convolutional neural networks dropout regularization application", Scientific Reports, 2022 7 words — < 1%  
Crossref

69 Xin Chen. "<![CDATA[A Modified PSO Structure Resulting in High Exploration Ability With Convergence Guaranteed]]>", IEEE Transactions on Systems Man and Cybernetics Part B (Cybernetics), 10/2007  
Crossref 7 words — < 1%

70 "Cuckoo Search and Firefly Algorithm", Springer Science and Business Media LLC, 2014  
Crossref 6 words — < 1%

71 "Recent Advances in Hybrid Metaheuristics for Data Clustering", Wiley, 2020  
Crossref 6 words — < 1%

72 Lecture Notes in Computer Science, 2006.  
Crossref 6 words — < 1%

73 Qizhen Li, Aijia Ouyang, Xuyu Peng, Xijun Hu. "QBGA-SVM for Face Recognition of livable cities", International Journal of Pattern Recognition and Artificial Intelligence, 2023  
Crossref 6 words — < 1%

74 repozitorij.unizg.hr  
Internet 6 words — < 1%

75 www.podaci.net  
Internet 6 words — < 1%

EXCLUDE QUOTES ON  
EXCLUDE BIBLIOGRAPHY ON

EXCLUDE SOURCES OFF  
EXCLUDE MATCHES OFF