

Data Clustering on Breast Cancer Data Using Firefly Algorithm with Golden Ratio Method

Murat DEMİR¹, Ali KARCI²

¹Muş Alparslan University, Vocational School, Muş, Turkey

²İnönü University, Faculty of Engineering, Department of Computer Engineering, Malatya, Turkey
m.demir@alparslan.edu.tr

Abstract—Heuristic methods are problem solving methods. In general, they obtain near-optimal solutions, and they do not take the care of provability of this case. The heuristic methods do not guarantee to obtain the optimal results; however, they guarantee to obtain near-optimal solutions in considerable time. In this paper, an application was performed by using firefly algorithm - one of the heuristic methods. The golden ratio was applied to different steps of firefly algorithm and different parameters of firefly algorithm to develop a new algorithm - called Firefly Algorithm with Golden Ratio (FAGR). It was shown that the golden ratio made firefly algorithm be superior to the firefly algorithm without golden ratio. At this aim, the developed algorithm was applied to WBCD database (breast cancer database) to cluster data obtained from breast cancer patients. The highest obtained success rate among all executions is 96% and the highest obtained average success rate in all executions is 94.5%.

Index Terms—artificial Intelligence, heuristic algorithms, clustering algorithms.

I. INTRODUCTION

The heuristic methods are derived from natural phenomena to satisfy any aim or reach to any goal. It is not possible to prove analytically that heuristic algorithms obtain optimum results in each execution. That is, heuristic algorithms converge to near-optimal solutions however; they do not guarantee to obtain optimum solutions. The properties of heuristic algorithms such as understandability, capability of heuristic algorithm to apply to indefinable exact solution for optimization problems analytically, their simplicity with respect to other algorithms, and their capability to apply to machine learning make them preferable among algorithms.

The meaning of optimization is the act of obtaining the best result under given circumstances. The optimization takes place in design, construction, and maintenance of any engineering system by taking some decisions at several stages of problem. In another way, optimization is to select a near-optimal or optimal solution in the alternative solutions. The optimization is the process of determining the conditions to obtain maximum or minimum. The optimization can be defined as the process of finding the conditions that give the maximum or minimum value of a function [1]. Any problem containing unknown parameter values under some constraints can be defined as optimization problems.

An optimization problem can be defined as follow.

Minimized or maximized $z = f(x), x = (x_1, x_2, \dots, x_n)^T$ (1)
 $z = f(x)$ is the optimized value. The optimization problem may be subject to some constraints or not.

The constraints are inequality ($g_i(x)$) and equality ($h_j(x)$) constraints.

$$\sum_{i=1}^n g_i(x) \leq 0 \text{ and } \sum_{j=1}^m h_j(x) = 0 \quad (2)$$

The domain intervals for parameters of firefly algorithm can be considered as constraints such as

$$\begin{aligned} 0 &\leq \gamma < \infty, \\ 0 &\leq \alpha \leq 1, \\ 0 &\leq \beta \leq 1, \\ 0 &\leq i_0 \leq 1 \end{aligned} \quad (3)$$

The underlined philosophy for robustness and strongness of heuristic methods can be summarized as follows: It is easy to apply these methods to optimization problems of different decision variables, constraints, and objective functions. They do not depend on the type of solution space, the number of decision variables, and the number of constraints. The well-defined mathematical knowledge for modelling system and objective function of problems is not required. They have simple and easy to use computation power without excessive computation complexities. They have efficacious solutions to the high-scale combinatorial and nonlinear problems [2]. They do not require the assumptions that are hard to be approved to adapt a solution algorithm to a given problem as done in classical algorithm [2]. They can be adapted to solve different types of problems, on contrary, classical algorithms require alteration on the interested problem. Finally, most of the computational heuristics algorithms are inspired by a natural event/process.

These advantages of computational heuristic methods made these methods are densely being used in many different fields such as computer engineering/science, bioinformatics, management science, engineering, etc. due to this case, so new versions of these methods have been proposed.

Most of the proposed methods are population-based computational intelligence methods, that is, they begin to search the solution with multiple candidate solutions. These methods can be categorized as physical, biological, social, chemical, social-biological, biological-geography, music and hybrid versions.

Physical based: Multi-point simulated annealing algorithm [3], Gravitational search algorithm [4], Electromagnetism-like algorithm [5, 6], Big bang-big crunch algorithm [7]. Biological based: Genetic algorithms [8,9], Ant colony algorithm [10], Bee colony algorithm [11, 12], Artificial immune algorithm [13], Firefly algorithm [14-

17], Saplings growing-up algorithm [18-23], Invasive weed optimization [24], Monkey search algorithm [25], Bacterial foraging algorithm [26], Cricket algorithm [27], Evolutionary Algorithms [28]. Social based: Multi-point tabu search algorithm [29], Imperialist competitive algorithm [30]. Chemical based: Artificial chemical reaction optimization algorithm [31], artificial atom algorithm [31-33]. Social-Biological based: Particle swarm optimization [34, 35], Cat swarm optimization [36]. Biological-Geography based: Biogeography-based algorithm [37]. Musical based: Harmony search algorithm [38]. Hybrid versions: Combining two or more computational heuristics algorithms [39].

II. FIREFLY ALGORITHM

The Firefly Algorithm was developed by the author Xin She Yang and it was based on the idealized behaviour of the flashing characteristics of fireflies [16]. The interesting behaviours of fireflies such as short and rhythmic flashes can be thought as operators of computational intelligence methods. These flashes are used as a communication tools by fireflies. The rate of flashing, rhythmic flash and the amount of time form part of the signal system that brings both sexes together [14, 15]. Assume that the distance of light source is d then the light intensity obeys the inverse square law (proportional to $1/d^2$) [15]. In another word, the light intensity I decrease as the distance r increases.

Besides, the combined factors make most fireflies visible for a limited distance such as several hundred meters at night, and this is enough for fireflies to communicate.

The flashing light can be used for formulation of objective function to be optimized. In the rest of this paper, we will first outline the basic formulation of the Firefly Algorithm (FA) and then discuss the implementation as well as its analysis in detail.

Yang idealized some of the flashing characteristics of fireflies to design a meta-heuristic algorithm. In order to describe Yang algorithm simply, the following three rules must be taken in care [14-16]:

1) All fireflies are attracted by other fireflies (they are unisex).

2) The brightness of light effect attractiveness (attractiveness is proportional to their brightness), this means that the firefly with less bright will move towards the brighter firefly.

3) The brightness of a firefly is affected or determined by the type of the objective function.

The basic steps of the firefly algorithm (FA) can be summarized as the pseudo code shown in Algorithm 1 based on the three rules given above [14, 15].

The details about firefly algorithm can be given as follow and all this information were obtained from Yang studies.

The light intensity $I(r)$ varies according to the inverse square law, and the light emitted by fireflies has different intensity depending on distance.

$$I(r) = \frac{I_s}{r^2} \quad (4)$$

where I_s is the intensity at the source. The medium light absorption coefficient γ affects the light intensity changes and

$$I = I_0 e^{-\gamma r} \quad (5)$$

where I_0 is the original light intensity. The light intensity affects the attractiveness of light seen by adjacent fireflies [14-16].

$$\beta = \beta_0 e^{-\gamma r^2} \quad (6)$$

The distance computations among fireflies are based on Euclidian distance, i.e. any two fireflies i and j at x_i and x_j , respectively, distance is [14, 15].

$$r_{ij} = |x_i - x_j| = \sqrt{\sum_{k=1}^d (x_{i,k} - x_{j,k})^2} \quad (7)$$

The movement of a firefly i with less attractiveness is attracted by a more attractive firefly j , and this is determined by equation [15, 16]

$$x_i = x_i + \beta_0 e^{-\gamma r_{ij}^2} (x_j - x_i) + \alpha \varepsilon_i \quad (8)$$

where ε_i is a vector of random numbers drawn from a uniform distribution or Gaussian distribution. Yang used this information for designing Firefly Algorithm (as seen in Algorithm 1).

Algorithm 1. Firefly Algorithm pseudo code

```

Begin
Objective function f(x) ,
X=(X1,X2,.....Xd)T
Generate initial population of fireflies
Xi (i=1,2,...,n)
Light intensity Ii at Xi is determined
by f(Xi)
Define light absorption coefficient γ
while (t<Maxgeneration)
    for i=1: n all n fireflies
        for j=1: i all n fireflies
            if (Ii > Ij)
                Move firefly i towards j in d-dimension
                via Levy flights
            end if
        end for
    end for
    Attractiveness varies with distance r
    via exp[-γr]
    Evaluate new solutions and update light
    intensity
end while
Rank the fireflies and find the current
best
Postprocess results and visulation
End

```

III. GOLDEN RATIO

Golden ratio is founded in morphological structures and shapes of many alive and lifeless. The most obvious examples of golden ratio are seen in the human body, leaves, tree branches, etc. The golden ratio in term of compliance and a numerical ratio of the geometric relation is a competent size in the mathematics and art. Any line can be divided in a point such that the ratio of small line segment over large line segment is equal to ratio of large line segment over whole line. Fig. 1 depicts this case [40].

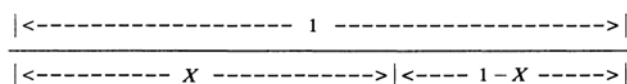


Figure 1. Golden ratio in a line segment [40]

Fig. 1 depicts a line segment of length 1 divided into two pieces.

$$(1/x) = x/(1-x) \text{ or } x^2 + x - 1 = 0 \quad (9)$$

The positive root of equation (9) is $x = -1 + \sqrt{5}/2 \approx 0.61803398875$. The ratio $1/x = 1 + \sqrt{5}/2 \approx 1.61803398875$

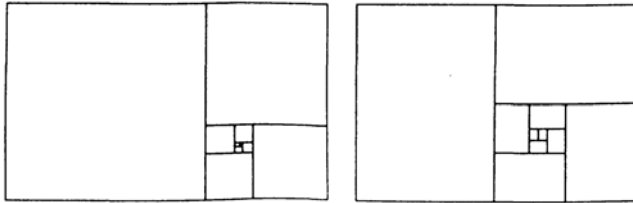


Figure 2. Spirals of rectangles [40]

If the short side of any rectangle over large side is equal to golden ratio, this rectangle is called golden rectangle as seen in Fig.2. Given an infinite number of nested rectangles created, and drawing several times squares in this rectangle and a circle of radius is equal to side of this squares concludes in golden spiral as seen in Fig.3.

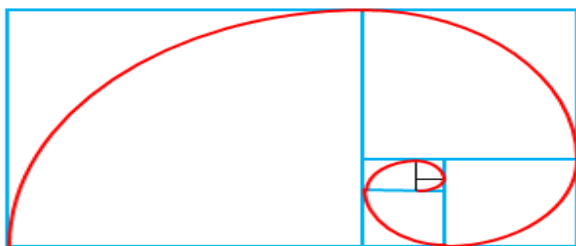


Figure 3. Golden rectangle and golden spiral

IV. FAGR - DEVELOPED APPLICATION

The data obtained from Wisconsin Diagnosis Breast Cancer Database was used in this study. The data obtained from database were formatted as seen in Table I.

TABLE I. THE REPRESENTATION OF KNOWLEDGE OBTAINED FROM DATABASE

1000025	5	1	1	1	2	1	3	1	1	2
1002945	5	4	4	5	7	10	3	2	1	2
1015425	3	1	1	1	2	2	3	1	1	2
1016277	6	8	8	1	3	4	3	7	1	2
1017023	4	1	1	3	2	1	3	1	1	2
1017122	2	1	2	1	2	1	3	1	1	4
1018099	4	2	1	1	2	1	2	1	1	2

The first column in Table I represents the code number, the last column represents whether tumour is benign or malignant where the value 2 means that tumour is benign and the value 4 means that tumour is malignant. The database contains 700 tuples and a little portion of this database contains N/A attributes. These tuples were removed and the remaining 578 tuples were used for training and 100 tuples were used for test. The code number is just a number and it was not used in the clustering

process. The process is not classification; it is a clustering process.

The attributes are the clump thickness, the uniformity of cell sizes, and uniformity of cell shapes, marginal adhesion, single epithelial cell size, bare nuclei, blend chromatin, normal nucleoli, and the mitoses (in Table II).

TABLE II. PRE-PROCESSED DATA IN DATABASE

5	1	1	1	2	1	3	1	1
5	4	4	5	7	10	3	2	1
3	1	1	1	2	2	3	1	1
6	8	8	1	3	4	3	7	1
4	1	1	3	2	1	3	1	1
2	1	2	1	2	1	3	1	1
4	2	1	1	2	1	2	1	1

There are 20 fireflies (there 20 matrices). Each firefly was represented by a matrix of sizes 9 columns, because database contains nine attributes. The output of developed software is matrix of sizes 1x9, and it represents the best results. Each firefly was compared with each tuple of database and a total error value was obtained. The aim is to minimize this error. The obtained best attribute values constitute the best matrix such as in Table III and Fig.4.

TABLE III. REPRESENTATION OF THE BEST MATRIX

best (1,1)	best (1,2)	best (1,3)	.	.	.	.	.	best (1,9)
---------------	---------------	---------------	---	---	---	---	---	---------------

The Manhattan distances between the best matrix values obtained by using developed firefly algorithm and the data remained for test are computed for determining differences. Then cluster are determined by using this differences.

best (1,1)	best (1,2)	best (1,3)	.	.	.	.	.	best (1,9)
↓	↓	↓	↓	↓	↓	↓	↓	↓
attribute1	.	.	.	.	.	.	attribute9	Manhattan distance
5	4	4	5	7	10	3	2	1
3	1	1	1	2	2	3	1	1
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
4	2	1	1	2	1	2	1	1

Figure 4. Determining clusters by using the best matrix. Figure

$m_1, m_2, \dots, m_n$ values represent the differences (Manhattan distances) between the best matrix values and corresponding data remained for test process, and these differences are used for determining cluster values. $m_i = |\text{best}(1,i) - \text{value of attribute } i|$

$$\begin{aligned} &\text{for } i \leftarrow 1, 2, \dots, n \\ &\quad \text{for } k \leftarrow 1, 2, \dots, 9 \\ &\quad m_i = \sum |(\text{best}(1,k) - (\text{value of attribute}_{i,k}))| \end{aligned}$$

A. Process of Developed Method

The differences between initial population and training data are obtained. The summation for any solution gives the error for the corresponding solution. These differences are Manhattan distances. The differences between training data and solution individuals are computed, and the differences for candidate solutions are summed up to obtain total difference (Fig.5).

```
for k←1,2,...,20
    for m←1,2,...,9

differences(k,m)=Σ(|population(k,m)-
training(:,m)|)
```

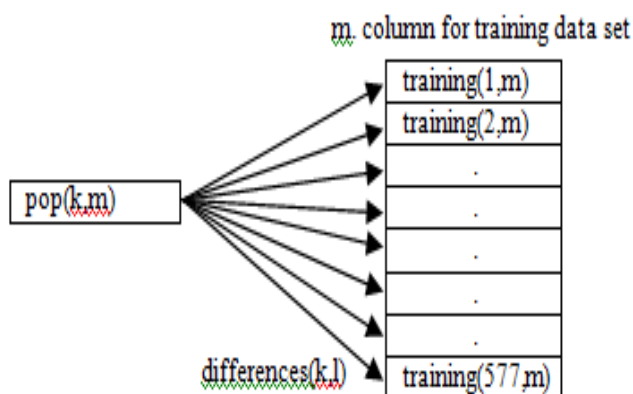


Figure 5. Differences of population values and training values

The above process is for finding the difference matrix (sumofdifferences) between population and training data (in Table IV).

```
for k←1,2,...,20
    sumofdifferences(1,k)=
Σ(differences(k,:))
```

TABLE IV. DIFFERENCES MATRICE

difference(1,1)	difference(1,2)	.	.	.	difference(1,9)
difference(2,1)	difference(2,2)	.	.	.	difference(2,9)
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
difference(20,1)	difference(20,2)	.	.	.	difference(20,9)
$\Sigma(\text{difference}(k,:))$					

If there is a total difference for each candidate solution, there will be 20 summed up differences (in Table V).

TABLE V. SUM OF DIFFERENCES MATRICE

Sum of difference(1,:)	Sum of difference(2,:)	.	.	.	Sum of difference(20,:)
---------------------------	---------------------------	---	---	---	----------------------------

These differences constitute the difference matrix (sumofdifferences). So, the difference of each firefly was

obtained in this way.

```
for k←1,2,...,20
    for m←1,2,...,20
        r(k,m)=sqrt((sumofdifferences
(1,k)- sumofdifferences (1,m)).^2)
```

r is an Euclidean distances amongst fireflies and these distances were assigned to r matrix.

```
for k←1,2,...,20
    sumofr(1,k)= Σ(r(k,:))
```

This code computes all distances between a specific firefly and the remaining fireflies, the obtained summand assigned to sumofr matrix. The light intensity and attractiveness were computed by using values in sumofr matrix.

```
for k←1,2,...,20
    i(1,k)= i0*(exp(-
gamma*sumofr(1,k)/1000000))
```

The matrix i is the matrix of light intensity, and the normalization was obtained by dividing all values to 1000000, because the limit of $I = I_0 e^{-r}$ closes to zero. The normalization must be done in an interval as done in this paper.

```
for k←1,2,...,20
    beta(1,k)=beta0*( exp(-
gamma*((sumofr(1,k)/1000000).^2))
```

The normalization was handled in the same way, while computing attractiveness.

After all these computation, next generation for candidate solution is obtained by using Eq.8.

The Manhattan distances are re-computed by using this new population (next generation) and the best matrix is also re-computed. These steps are repeated until a termination criterion is met. When algorithm is terminated, the best matrix represents the solution.

Attractiveness evaluate steps in firefly algorithm

```
for j=1:i all n fireflies
    if(Ij > Ii)
        Move firefly i towards j in
        d-dimension via Levy flights
    end if
    Attractiveness varies with distance
    r via exp[-γr]
    Evaluate new solutions and update
    Light intensity
```

The attractiveness and light intensity values were used in the remaining steps of firefly algorithm. The solution values were computed and inserted into population. The minimum value in the sumofdifferences matrix was regarded as minimum value, and if fewer values were obtained in the subsequent steps, this value was regarded as new minimum value. This minimum value depicts the solution in the population. This value is a part of best matrix. The subsequent steps demonstrated the result of each execution. Success rate is an accuracy value percentage for clustering of this data set.

Firefly Algorithm (FA) includes the parameters such as α , β , γ and i_0 . α , β and i_0 are in closed interval $[0,1]$; γ is in semi-closed interval $[0, \infty)$.

The proposed method obtained the best results when parameters in $[0, 1]$ have fractional part of golden ratio (0.618). Due to γ is in interval $[0, \infty)$, the best result is obtained for $\gamma = 2.1$. Although this case, in the closed interval $[0, 1]$, the best result was obtained for $\gamma = 0.618$.

Ff1: The golden ratio was not applied in this case, so all parameters are equal to 1. $\gamma=1$, $\beta_0=1$, $\alpha=1$ and $i_0=1$, the results of execution were illustrated in Table VI for random initial population.

TABLE VI. RESULTS OF Ff1

best (1,1)	best (1,2)	best (1,3)	best (1,4)	best (1,5)	best (1,6)	best (1,7)	best (1,8)	best (1,9)	Accuracy rate (%)
8	7	7	8	7	8	7	8	7	95
1	2	1	1	2	1	2	2	1	93
8	8	8	8	8	8	8	8	8	95
1	2	3	1	3	2	1	3	1	91
3	2	2	2	2	2	3	3	2	91
6	6	8	8	6	8	6	7	6	94
1	1	1	1	3	1	3	1	3	94
1	3	3	1	3	1	3	3	1	94
8	8	8	9	9	8	8	9	9	94
6	6	6	8	6	8	6	8	6	93
Average=93,4									

The content of the best matrix is as follow after the first step of Ff1.

8	7	7	8	7	8	7	8	7
---	---	---	---	---	---	---	---	---

Some Manhattan distances are seen in Table 7 by applying the results of this step. By this way, when Manhattan distances for all test data are obtained, the cluster of each test data can also be determined.

TABLE VII. FOR AN EXAMPLE MANHATTAN DISTANCE VALUES FOR Ff1 EXPERIMENT1

att. 1	att. 2	att. 3	att. 4	att. 5	att. 6	att. 7	att. 8	att. 9	m _j
8	4	4	5	4	7	7	8	2	18
5	4	5	1	8	1	3	6	1	35
3	3	2	6	3	3	3	5	1	38
3	1	1	3	8	1	5	8	1	38
5	1	2	10	4	5	2	1	1	40
5	2	2	2	2	2	3	2	2	45
5	3	6	1	2	1	1	1	1	46
6	8	7	8	6	8	8	9	1	12
6	8	7	5	6	8	8	9	2	14
10	5	6	10	6	10	7	7	10	14
8	10	10	10	5	10	8	10	6	16
8	10	10	8	7	10	9	7	1	17
8	10	10	8	6	9	3	10	10	17
6	10	10	10	8	10	10	10	7	18

The Manhattan distances for two sets are distinguishable except some errors as seen in Table VII. This depicts that the best matrix obtained by using FA can cluster data in a better way.

Ff2: The golden ratio was applied to gamma parameter of firefly algorithm at this experiment, and the fractional part of golden ratio was selected for gamma parameter. $\gamma=0.618$, $\beta_0=1$, $\alpha=1$ and $i_0=1$, the results of execution were illustrated in Table VII for random initial population.

Ff3: The golden ratio was not applied in this experiment; that is, all parameters of Firefly algorithm were determined randomly, i.e. $\gamma=0.2$. $\gamma=0.2$, $\beta_0=1$, $\alpha=1$ and $i_0=1$, the results of execution were illustrated in Table VII for random initial population.

Ff4: The golden ratio was not applied in this experiment; that is, all parameters of Firefly algorithm were determined randomly, i.e. $\gamma=0.95$. $\gamma=0.95$, $\beta_0=1$, $\alpha=1$ and $i_0=1$, the results of execution were illustrated in Table VII for random initial population.

Ff5: The parameter gamma of Firefly algorithm was selected as golden ratio such as $\gamma=1.618$. $\gamma=1.618$, $\beta_0=1$, $\alpha=1$ and $i_0=1$, the results of execution were illustrated in Table VII for random initial population.

Ff6: The golden ratio was not applied in this experiment; that is, all parameters of Firefly algorithm were determined randomly, i.e. $\gamma=2.1$. $\gamma=2.1$, $\beta_0=1$, $\alpha=1$ and $i_0=1$, the results of execution were illustrated in Table VIII for random initial population.

TABLE VIII. SUCCESS RATES

	Ff1	Ff2	Ff3	Ff4	Ff5	Ff6
Experiment1	%95	%95	%95	%95	%92	%96
Experiment2	%93	%92	%95	%88	%95	%96
Experiment3	%95	%93	%85	%93	%95	%94
Experiment4	%91	%95	%91	%93	%93	%94
Experiment5	%91	%95	%85	%92	%90	%93
Experiment6	%94	%93	%89	%94	%96	%95
Experiment7	%94	%95	%85	%95	%93	%93
Experiment8	%94	%92	%95	%95	%95	%96
Experiment9	%94	%95	%95	%89	%92	%94
Experiment10	%93	%95	%95	%91	%96	%95
Average	%93,4	%94	%91	%92,5	%93,7	%94,6

The gamma parameter can be determined in semi-closed interval $[0, \infty)$ by designer for firefly algorithm. Due to this case, execution got the largest value in execution 6 for $\gamma=2.1$.

While executions 1, 2, 3, 4, 5, and 6 are taken in care carefully, it can be seen that algorithm obtained the highest value for $\gamma=0.618$ in closed interval $[0, 1]$ where 0.618 is fraction part of golden ratio (Fig. 6).

The success rate appears to decrease immediately while gamma value goes far away from the fraction part of golden ratio 0.618.



Figure 6. The figure of success rates

The effects of changes of the second parameter can be seen in the following four experimental results.

Ff7: The parameters beta and gamma were selected as fractional part of golden ratio. $\gamma=0.618$, $\beta_0=0.618$, $\alpha=1$ and $i_0=1$, the results of execution were illustrated in Table VIII for random initial population.

Ff8: The value of gamma parameter was selected as fractional part of golden ratio and beta was selected as 0.2. Gamma=0.618, beta₀=0.2, alpha=1 and i₀=1, the results of execution were illustrated in Table VIII for random initial population.

Ff9: At this experiment, gamma were selected as fractional part of golden ratio and beta as golden ratio. Gamma=0.618, beta₀=1.618, alpha=1 and i₀=1, the results of execution were illustrated in Table VIII for random initial population.

Ff10: At this experiment, gamma was selected as fractional part of golden ratio and beta was selected different from golden ratio. Gamma=0.618, beta₀=3, alpha=1 and i₀=1, the results of execution were illustrated in Table IX for random initial population.

TABLE IX. SUCCESS RATES

	Ff7	Ff8	Ff9	Ff10
Experiment1	%95	%94	%93	%93
Experiment2	%95	%94	%94	%95
Experiment3	%94	%94	%91	%90
Experiment4	%92	%95	%92	%95
Experiment5	%93	%93	%96	%93
Experiment6	%94	%92	%90	%90
Experiment7	%95	%93	%95	%95
Experiment8	%94	%94	%91	%95
Experiment9	%93	%92	%94	%91
Experiment10	%95	%93	%94	%95
Average	%94	%93,4	%93	%93,2

The success rate reached the highest value for executions 7, 8, 9 and 10 in case of beta₀=0.618. This value is the fraction of golden ratio (Fig. 7).

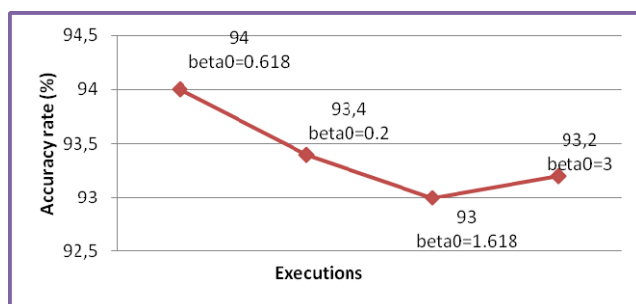


Figure 7. Figure of success rates

Ff11: The three parameters gamma, beta and alpha were selected as the fractional part of golden ratio. Gamma=0.618, beta₀=0.618, alpha=0.618 and i₀=1, the results of execution were illustrated in Table IX for random initial population.

Ff12: The gamma and beta were selected as the fractional part of golden ratio and alpha as 0.25. Gamma=0.618, beta₀=0.618, alpha=0.25 and i₀=1; the results of execution were illustrated in Table IX for random initial population.

Ff13: The gamma and beta were selected as the fractional part of golden ratio and alpha as 0.88. Gamma=0.618, beta₀=0.618, alpha=0.88 and i₀=1, the results of execution were illustrated in Table X for random initial population.

TABLE X. SUCCESS RATES

	Ff11	Ff12	Ff13
Experiment1	%95	%90	%92
Experiment2	%93	%92	%94
Experiment3	%95	%88	%95
Experiment4	%94	%80	%95
Experiment5	%92	%82	%95
Experiment6	%94	%84	%90
Experiment7	%94	%84	%95
Experiment8	%94	%83	%94
Experiment9	%95	%84	%91
Experiment10	%95	%86	%94
Average	%94,1	%85,3	%93,5

The success rate reached the highest value in executions 11, 12 and 13 for alpha=0.618 and execution 11. This case is illustrated in Fig. 8.

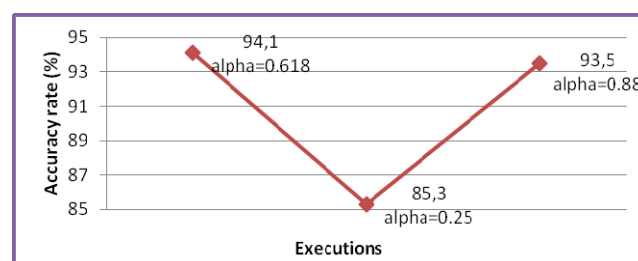


Figure 8. The figure of success rates

Ff14: The gamma, beta, alpha and i₀ were selected as the fractional part of golden ratio. Gamma=0.618, beta₀=0.618, alpha=0.618 and i₀=0.618, the results of execution were illustrated in Table X for random initial population.

Ff15: The gamma, beta and alpha were selected as the fractional part of golden ratio, and i₀ was selected different from golden ratio such as 0.2. Gamma=0.618, beta₀=0.618, alpha=0.618 and i₀=0.2, the results of execution were illustrated in Table X for random initial population.

Ff16: The gamma, beta and alpha were selected as the fractional part of golden ratio, and i₀ was selected different from golden ratio such as 0.9. Gamma=0.618, beta₀=0.618, alpha=0.618 and i₀=0.9, the results of execution were illustrated in Table XI for random initial population.

When the success rates for executions 14,15 and 16 are investigated in care, the success rate reached the better value for execution 14 and parameter i₀=0.618 (Fig. 9).

TABLE XI. SUCCESS RATES

	Ff14	Ff15	Ff16
Experiment1	%93	%94	%94
Experiment2	%95	%90	%92
Experiment3	%92	%93	%91
Experiment4	%95	%95	%94
Experiment5	%93	%94	%93
Experiment6	%94	%91	%96
Experiment7	%94	%95	%95
Experiment8	%96	%91	%93
Experiment9	%94	%94	%91
Experiment10	%94	%95	%91
Average	%94	%93,2	%93

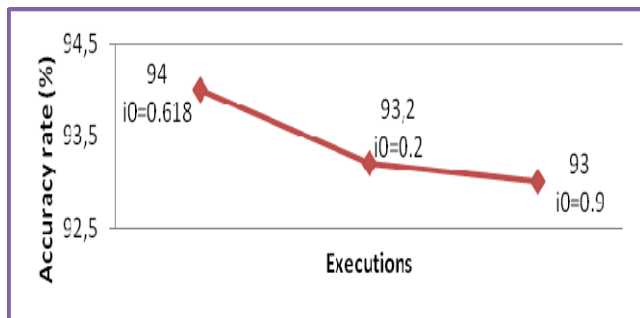


Figure 9. The figure of success rates

The initial populations for the remaining executions are not random, and they were generated by using uniform initial population as in Sapling Growing-up Algorithm.

Sapling Growing-up algorithm which is superior to other optimization problems in regular population advantage for one of the parties, creating a range of possible solutions, and the smallest and largest values in a population of them to create and distribute in the best way as possible, so that the solution interval.

When this is satisfied, the best solution will be obtained in shorter time and better success rate. The meta-heuristic algorithms may get stuck in local solution or diverged from the best solution. Sapling Growing-up Algorithm can get rid of these problems by using uniform initial population [18-23].

TABLE XII. GENERATING INITIAL POPULATION FOR SAPLING GROWING-UP ALGORITHM

Min1	Min2	Min3	Min4
$L_i + (u_i - l_i) * r$	$L_i + (u_i - l_i) * r$	$L_i + (u_i - l_i) * r$	$L_i + (u_i - l_i) * (1-r)$
$L_i + (u_i - l_i) * r$	$L_i + (u_i - l_i) * r$	$L_i + (u_i - l_i) * (1-r)$	$L_i + (u_i - l_i) * r$
$L_i + (u_i - l_i) * r$	$L_i + (u_i - l_i) * r$	$L_i + (u_i - l_i) * (1-r)$	$L_i + (u_i - l_i) * (1-r)$
$L_i + (u_i - l_i) * r$	$L_i + (u_i - l_i) * (1-r)$	$L_i + (u_i - l_i) * r$	$L_i + (u_i - l_i) * r$
$L_i + (u_i - l_i) * r$	$L_i + (u_i - l_i) * (1-r)$	$L_i + (u_i - l_i) * r$	$L_i + (u_i - l_i) * (1-r)$
$L_i + (u_i - l_i) * r$	$L_i + (u_i - l_i) * (1-r)$	$L_i + (u_i - l_i) * (1-r)$	$L_i + (u_i - l_i) * r$
$L_i + (u_i - l_i) * r$	$L_i + (u_i - l_i) * (1-r)$	$L_i + (u_i - l_i) * (1-r)$	$L_i + (u_i - l_i) * (1-r)$
$L_i + (u_i - l_i) * (1-r)$	$L_i + (u_i - l_i) * r$	$L_i + (u_i - l_i) * r$	$L_i + (u_i - l_i) * r$
$L_i + (u_i - l_i) * (1-r)$	$L_i + (u_i - l_i) * r$	$L_i + (u_i - l_i) * r$	$L_i + (u_i - l_i) * r$
$L_i + (u_i - l_i) * (1-r)$	$L_i + (u_i - l_i) * r$	$L_i + (u_i - l_i) * (1-r)$	$L_i + (u_i - l_i) * r$
$L_i + (u_i - l_i) * (1-r)$	$L_i + (u_i - l_i) * r$	$L_i + (u_i - l_i) * r$	$L_i + (u_i - l_i) * (1-r)$
$L_i + (u_i - l_i) * (1-r)$	$L_i + (u_i - l_i) * (1-r)$	$L_i + (u_i - l_i) * r$	$L_i + (u_i - l_i) * r$
$L_i + (u_i - l_i) * (1-r)$	$L_i + (u_i - l_i) * (1-r)$	$L_i + (u_i - l_i) * (1-r)$	$L_i + (u_i - l_i) * r$
Max1	Max2	Max3	Max4

Table XII depicts for attributes and how initial population is generated for these attributes. When the number of attributes increases, the initial population size will increase (i.e. for 9 attributes $2^9=512$). In this case, attributes can be grouped and the size of initial population will be decreased in this way.

TABLE XIII. THE REPRESENTATION OF GROUPED INITIAL POPULATION

$L_i + (u_i - l_i) * r$	$L_i + (u_i - l_i) * r$	$L_i + (u_i - l_i) * r$	$L_i + (u_i - l_i) * r$	$L_i + (u_i - l_i) * r$	$L_i + (u_i - l_i) * r$	$L_i + (u_i - l_i) * (1-r)$	$L_i + (u_i - l_i) * (1-r)$	$L_i + (u_i - l_i) * (1-r)$
Group1		Group2		Group3		Group4		

Each group is multiplied by r or $(1-r)$ and initial firefly values are obtained in this way. By the way, a population of size 20 was generated and each firefly contains 9 attributes (in Table XIII).

L_i represent lower bound values in the training data and U_i represent upper bound values in the training data. When initial population is created by using minimum and maximum values, the convergence to optimal or near-optimal value will be better.

New x matrices were obtained in this manner. The subsequent executions used uniform initial population as in Sapling Growing-up Algorithm.

The obtained results will be given in the following executions. The initial populations for executions 17, 18, 19, 20, 21, 22, and 23 are uniform initial populations.

Ff17: The initial population was generated by using uniform initial population method as seen in Sapling Growing-up Algorithm.

The maximum and minimum values for each attribute were determined by using data in database. The largest value in a column was determined as maximum value and the lower value in a column was determined as minimum value for corresponding attribute.

$\gamma=0.618$, $\beta_0=0.618$, $\alpha=0.618$ and $i_0=0.618$, initial population was generated by using uniform initial population technique as in Sapling Growing-up Algorithm, and results were illustrated in Table XIII.

Ff18: The initial population was generated by using uniform initial population method as seen in Sapling Growing-up Algorithm.

The maximum and minimum values for each attribute were determined by using data in database. The largest value in a column was determined as maximum value and the lower value in a column was determined as minimum value for corresponding attribute.

$\gamma=0.1$, $\beta_0=0.1$, $\alpha=0.1$ and $i_0=0.1$, initial population was generated by using uniform initial population technique as in Sapling Growing-up Algorithm, and results were illustrated in Table XIII.

Ff19: The initial population was generated by using uniform initial population method as seen in Sapling Growing-up Algorithm.

The maximum and minimum values for each attribute were determined by using data in database. The largest

value in a column was determined as maximum value and the lower value in a column was determined as minimum value for corresponding attribute.

$\gamma=0.4$, $\beta_0=0.4$, $\alpha=0.4$ and $i_0=0.4$, initial population was generated by using uniform initial population technique as in Sapling Growing-up Algorithm, and results were illustrated in Table XIII.

Ff20: The initial population was generated by using uniform initial population method as seen in Sapling Growing-up Algorithm.

The maximum and minimum values for each attribute were determined by using data in database. The largest value in a column was determined as maximum value and the lower value in a column was determined as minimum value for corresponding attribute.

$\gamma=0.85$, $\beta_0=0.85$, $\alpha=0.85$ and $i_0=0.85$, initial population was generated by using uniform initial population technique as in Sapling Growing-up Algorithm, and results were illustrated in Table XIV.

TABLE XIV. SUCCESS RATES

	Ff17	Ff18	Ff19	Ff20
Experiment1	%95	%84	%91	%93
Experiment2	%95	%82	%89	%90
Experiment3	%94	%89	%89	%95
Experiment4	%95	%91	%84	%94
Experiment5	%95	%85	%88	%90
Experiment6	%95	%89	%84	%92
Experiment7	%96	%89	%86	%95
Experiment8	%95	%84	%84	%95
Experiment9	%95	%89	%77	%95
Experiment10	%94	%77	%89	%93
Average	%94,9	%85,9	%86,1	%93,2

When results of executions 17, 18, 19 and 20 were investigated carefully, all parameter values are 0.618 and the better success rate was obtained in execution 17 (Fig. 10).

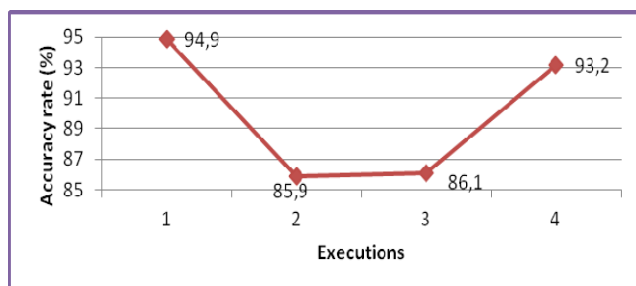


Figure 10. The success rates

The success rates are illustrated as seen in Fig. 10 and success rate reached the highest rate value of all the trials. The better success rate is 94.9%.

In addition, results previously obtained with the 96% rate of these can be found here. This step is the superior step in all executions, since the highest success rate and the highest average success rate were obtained in this step. The effects of uniform initial population and golden ratio can be observed here.

Ff21: The initial population was generated by using

uniform initial population method as seen in Sapling Growing-up Algorithm.

The maximum and minimum values for all attributes were determined by using data in database. The largest value in a column was determined as maximum value and the lower value in a column was determined as minimum value for corresponding attribute.

$\gamma=0.618$, $\beta_0=0.618$, $\alpha=0.618$ and $i_0=0.618$, initial population was generated by using uniform initial population technique as in Sapling Growing-up Algorithm, and results were illustrated in Table XIV.

Ff22: The initial population was generated by using uniform initial population method as seen in Sapling Growing-up Algorithm.

The maximum and minimum values for all attributes were determined by using data in database. The largest value in a column was determined as maximum value and the lower value in a column was determined as minimum value for corresponding attribute.

$\gamma=0.3$, $\beta_0=0.3$, $\alpha=0.3$ and $i_0=0.3$, initial population was generated by using uniform initial population technique as in Sapling Growing-up Algorithm, and results were illustrated in Table XIV.

Ff23: The initial population was generated by using uniform initial population method as seen in Sapling Growing-up Algorithm.

The maximum and minimum values for all attributes were determined by using data in database. The largest value in a column was determined as maximum value and the lower value in a column was determined as minimum value for corresponding attribute.

$\gamma=0.9$, $\beta_0=0.9$, $\alpha=0.9$ and $i_0=0.9$, initial population was generated by using uniform initial population technique as in Sapling Growing-up Algorithm, and results were illustrated in Table XV and Fig. 11.

TABLE XV. SUCCESS RATES

	Ff21	Ff22	Ff23
Experiment1	%93	%89	%92
Experiment2	%93	%84	%93
Experiment3	%95	%81	%93
Experiment4	%95	%89	%93
Experiment5	%95	%84	%89
Experiment6	%93	%82	%95
Experiment7	%94	%89	%94
Experiment8	%95	%89	%87
Experiment9	%90	%85	%95
Experiment10	%92	%86	%94
Average	%93,5	%85,8	%92,5

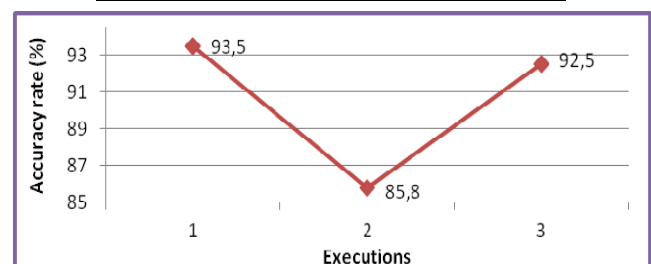


Figure 11. The success rates

V. CONCLUSIONS

In this study, the firefly algorithm was applied to clustering problems successfully. There is only one study about firefly algorithm and golden ratio. In this study [41], the Cartesian distance between the best firefly and the worst firefly was divided to inverse of golden ratio, and a dx value was obtained and this value was used in this study. dx is the ratio of difference between two points and value of ϕ . This means that $dx = |x_i - x_j| / \phi$.

There is not any other study about firefly algorithm and golden ratio. In this respect, the study will shed light on this matter structure. The database used in this study was used in many other studies. The methods and obtained results can be given as follows. However, these studies classified the data in the database. We clustered the data without using "benign, malignant".

- The study involved artificial neural networks applied to this database obtained success rate 100%. However, 35 tuples were used for training and 34 tuples were used for testing. In total, they used 69 data instead of 699 data [42].
- In another study, Artificial Neural Networks, Multilayer Perceptron, Radial Basis Function Networks, Self-Organizing Feature Maps, Vector Quantized Learning methods were applied to this database and the success rates are 95.74%, 96.18%, 98.5 and 96.7% respectively. 341 tuples in this database were used in this study [43].
- The Nearest Neighbour Method, Naive Bayes Classifier, Back Propagation Neural Networks and Support Vector Machine were applied to this database and obtained results are 96.36%, 95.60%, 96.09% and 96.56% respectively [44].
- Principal Component Analysis Fuzzy Neural Networks and (PCA-FNN) and Principal Component Analysis Artificial Neural Networks (PCA-ANN) were applied to this database and obtained success rates are 95.3% and 97.2% respectively [45].

Most of meta-heuristic methods have parameters and the values of these parameters are determined randomly in general. The golden ratio can be applied to most of these parameters. For this purpose, the golden ratio application to these parameters is an open problem.

The golden ratio was applied to parameters of firefly algorithm and the satisfaction results were obtained. The initial population is generated randomly in the most of the heuristic algorithms.

Another important contribution of this paper is to generate initial population for firefly algorithm by using uniform initial population method and this population made success rate be better.

The best obtained results were obtained in the execution 17 which contains effect of golden ratio and uniform initial population.

REFERENCES

- [1] K.G. Murty, "Optimization Models For Decision Making", Internet Edition, Models for Decision Making, vol 1, Chapter 1, pp. 1-8, 2003.
- [2] B. Alataş, "ACROA: Artificial Chemical Reaction Optimization Algorithm for Global Optimization", Expert Systems with Applications, vol 38, pp. 13170-13180, 2011, doi: 10.1016/j.eswa.2011.04.126.
- [3] L. Lamberti, C. Pappalè, "Weight optimization of skeletal structures with multi-point simulated annealing", Computer Modelling in Engineering and Sciences, vol. 18, no. 3, pp. 183-221, 2007, doi:10.3970/cmes.2007.018.183.
- [4] R.-E. Precup, R.-C. David, E. M. Petriu, S. Preitl, A. S. Paul, "Gravitational search algorithm-based tuning of fuzzy control systems with a reduced parametric sensitivity", in Soft Computing in Industrial Applications, A. Gaspar-Cunha, R. Takahashi, G. Schaefer, and L. Costa, Eds., Springer-Verlag, Berlin, Heidelberg, Advances in Intelligent and Soft Computing, vol. 96, pp. 141-150, 2011, doi:10.1007/978-3-642-20505-7_12.
- [5] S.I. Birbil, S.C. Fang, "An electromagnetism-like mechanism for global optimization", Journal of Global Optimization, vol 25, pp. 263-282, 2003, doi: 10.1023/A: 1022452626305.
- [6] R. Özdağ, A. Karci, "The Application of Electromagnetism-like Algorithm for the Dynamic Deployment Problem in Wireless Sensor Networks", in Proc. 2nd International Eurasian Conference on Mathematical Sciences and Applications, Sarajevo, Bosnia and Herzegovina, Aug. 26-29, 2013, pp. 199.
- [7] O.K. Erol, İ. Eksin, "A new optimization method: Big bang-big crunch", Advances in Engineering Software, vol 37, no. 2, pp. 106-111, February 2006, doi:10.1016/j.advengsoft.2005.04.005.
- [8] J.-T. Tsai, "Solving Japanese nongrams by Taguch-based genetic algorithm", Applied Intelligence, vol 37, no. 3, pp. 405-419, 2012, doi: 10.1007/s10489-011-0335-7.
- [9] H. Xing, R. Qu, "A compact genetic algorithm for the network coding based resource minimization problem", Applied Intelligence, vol 36, no. 4, pp. 809-823, 2011, doi:10.1007/s10489-011-0298-8.
- [10] J. Rivero, D. Coadra, J. Calle, P. Isasi, "Using the ACO algorithm for path searches in social networks", Applied Intelligence, vol 36, no. 4, pp. 899-917, 2011, doi: 10.1007/s10489-011-0304-1.
- [11] D. Karaboğa, B. Baştürk, "A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm", Journal of Global Optimization, vol 39, no. 3, pp. 459-471, 2007, doi: 10.1007/s10898-007-9149-x.
- [12] B. Akay, D. Karaboğa, "A Modified Artificial Bee Colony Algorithm for Real-Parameter Optimization", Information Sciences, vol 192, no. 1, pp. 120-142, 2012, doi:10.1016/j.ins.2010.07.015.
- [13] L.N. De Castro, F.J. Von Zuben, "Learning and optimization using the clonal selection principle", IEEE Transactions on Evolutionary Computation, vol 6, no. 3, pp. 239-251, 2002, doi: 10.1109/TEVC.2002.1011539.
- [14] X.-S. Yang, "Firefly algorithm, Levy flights and global optimization", in Proc. Research and Development in Intelligent Systems XXVI (Eds M. Bramer, R. Ellis, M. Petridis), Springer London, 2010, pp. 209-218, doi: 10.1007/978-1-84882-983-1_15.
- [15] X.-S. Yang, "Firefly algorithms for multimodal optimization", Stochastic Algorithms: Foundations and Applications, Lecture Notes in Computer Science, Springer-Verlag, Berlin, vol 5792, pp. 169-178, 2009, doi:10.1007/978-3-642-04944-6_14.
- [16] X.-S. Yang, "Firefly algorithm, stochastic test functions and design optimisation", International Journal of Bio-Inspired Computation, vol 2, no. 2, pp. 78-84, 2010, doi:10.1504/IJBIC.2010.032124.
- [17] X.-S. Yang, "Harmony Search as a Metaheuristic Algorithm", Music-Inspired Harmony Search Algorithm: Theory and Applications, Studies in Computational Intelligence, Springer Berlin, vol. 191, pp. 1-14, 2009, doi:10.1007/978-3-642-00185-7_1.
- [18] A. Karci, "Theory of saplings growing-up algorithm", in Proc. ICANNGA-2007: Adaptive and Natural Computing Algorithms, Editors: Bartłomiej Beliczynski, Andrej Dzielinski, Marcin Iwanowski, Bernardete Ribeiro, Berlin Heidelberg, LNCS, vol 4431, pp 450-460, 2007, doi: 10.1007/978-3-540-71618-1_50.
- [19] A. Karci, Bilal Alataş, "Thinking Capability of Saplings Growing Up Algorithm", in Proc. IDEAL-2006: 7th International Conference on Intelligent Data Engineering and Automated Learning, LNCS, vol 4224, 2006, pp.386-393, doi:10.1007/11875581_47.
- [20] A. Karci, "Saplings Sowing and Growing up Algorithm Convergence Properties", in Proc. INISTA-2007: International Symposium on Innovations in Intelligent Systems and Applications, Yıldız Technical University, İstanbul, 2007, pp. 322-326.
- [21] A. Karci, M. Yiğiter, M. Demir, "Natural Inspired Computational Intelligence Method: Saplings Growing Up Algorithm", in Proc. Ikecco'2007 International Kyrgyz-Kazak Electronics And Computer Conference, Bishkek-Almaty, 2007, pp.1-8.
- [22] M. Demir, M. Yiğiter, A. Karci, "Application of Saplings Growing Up Algorithm to Clustering Medical Data", in Proc. Ikecco'2007 International Kyrgyz-Kazak Electronics And Computer Conference, Bishkek-Almaty, 2007, pp.9-15.

- [23] M. Demir, A. Karcı, M. Özdemir, "Fidan Gelişim Algoritması Yardımı ile DNA Motiflerinin Keşfi", Çankaya University Journal of Science and Engineering, volume 8, no. 1, pp. 51-62, 2011.
- [24] A.R. Mehrabian, C. Lucas, "A novel numerical optimization algorithm inspired from weed colonization", Ecological Informatics, vol 1, no. 4, pp. 355-366, 2006, doi:10.1016/j.ecoinf.2006.07.003.
- [25] A. Mucherino, O. Seref, "Monkey search: A novel metaheuristics search for global optimization I. Continuous parameter optimization", Evolutionary Computation, vol 953, no. 1, pp. 25-49, 2007, doi: 10.1063/1.2817338.
- [26] K.M. Passino, "Biomimicry of bacterial foraging for distributed optimization and control", IEEE Control Systems Magazine, vol 22, no. 3, pp. 52-67, 2002, doi: 10.1109/MCS.2002.1004010.
- [27] M. Canayaz, A. Karcı, "A New Metaheuristic Cricket-Inspired Algorithm", in Proc. 2nd International Eurasain Conference on Mathematical Sciences and Applications, Sarajevo, Bosnia and Hercegovina, Aug. 26-29, 2013, pp. 176.
- [28] E. Deniz Ülker, A. Haydar, "Comparing the Robustness of Evolutionary Algorithms on the Basis of Benchmark Functions", Advances in Electrical and Computer Engineering, vol. 13, no. 2, pp. 59-64, 2013, doi:10.4316/AECE.2013.02010.
- [29] D. Niizuma, K. Yasuda, A. Ishigame, "Multi-point tabu search for traveling salesman problems", IEEE Transactions on Electrical and Electronic Engineering, vol 1, no. 1, pp. 126-129, 2006, doi:10.1002/tee.20028.
- [30] E.A. Gargari, C. Lucas, "Imperialist competitive algorithm: An algorithm for optimization inspired by imperialistic competition", in Proc. IEEE congress on evolutionary computation, Singapore, 2007, pp. 4661-4667, doi: 10.1109/CEC.2007.4425083.
- [31] A. Karcı, "A new Metaheuristic Algorithm Based Chemical Process: Atom Algorithm", in Proc. 1st International Eurasain Conference on Mathematical Sciences and Applications, Prishtine, Kosovo, Sep. 3-7, 2012, pp. 83-84.
- [32] A.Erdoğan Yıldırım, A. Karcı, "Solutions of Travelling Salesman Problem Using Genetic Algorithm and Atom Algorithm", in Proc. 2nd International Eurasain Conference on Mathematical Sciences and Applications, Sarajevo, Bosnia and Hercegovina, Aug. 26-29, 2013, pp. 134.
- [33] A. Karadoğan, A. Karcı, "Artificial Atom Algorithm for Reinforcement Learning", in Proc. 2nd International Eurasain Conference on Mathematical Sciences and Applications, Sarajevo, Bosnia and Hercegovina, Aug. 26-29, 2013, pp. 379.
- [34] J. Kennedy, R.C. Eberhart, "Particle swarm optimization", in Proc. of IEEE international conference on neural Networks, Australia, 1995, vol 4, pp. 1942-1948, doi: 10.1109/ICNN.1995.488968.
- [35] N. A. El-Hefnawy, "Solving Bi-level Problems Using Modified Particle Swarm Optimization Algorithm", International Journal of Artificial Intelligence, vol. 12, no. 2, pp. 88-101, 2014.
- [36] S.C. Chu, P.W. Tsai, J.S. Pan, "Cat swarm optimization", PRICAI 2006: Trends in Artificial Intelligence Lecture Notes in Computer Science, Volume 4099, pp 854-858, 2006, doi: 10.1007/978-3-540-36668-3_94.
- [37] D. Simon, "Biogeography-based optimization", IEEE Transactions on Evolutionary Computation, vol 12, no. 6, pp. 702-713, 2008, doi: 10.1109/TEVC.2008.919004.
- [38] K.S. Lee, Z.W. Geem, "A new metaheuristics algorithm for continues engineering optimization: Harmony search theory and practice", Computer Methods in Applied Mechanics and Engineering, vol 194, no. 36-38, pp. 3902-3933, 2005, doi: 10.1016/j.cma.2004.09.007.
- [39] F. Valdez, P. Malin, O. Castillo, "An improved evolutionary method with fuzzy logic for combining Particle Swarm Optimization and Genetic Algorithms", Applied Soft Computing, vol. 11, no. 2, pp. 2625-2632, 2011, doi:10.1016/j.asoc.2010.10.010.
- [40] G. Markowsky, "Misconceptions about the Golden Ratio", The College Mathematics Journal, Vol. 23, No. 1, pp. 2-19, 1992.
- [41] G. Wang, L. Guo, H. Duan, L. Liu and H. Wang, "A Modified Firefly Algorithm for UCAV Path Planning", International Journal of Hybrid Information Technology, vol. 5, no. 3, pp. 123-144, 3 July, 2012.
- [42] A. E. Temiz "Determination Of Breast Cancer Using ANN", Electronic Letters on Science & Engineering, vol. 3, no. 2, pp. 15-20, 2007.
- [43] T. Kıyan, T. Yıldırım, "Eğitici ve Eğiticişiz Nöral Algoritmalar Kullanarak Göğüs Kanseri Teşhisi", in Proc. Elektrik -Elektronik -Bilgisayar Mühendisliği 10. Ulusal Kongresi, İstanbul, 2003, pp. 453-456.
- [44] A. Eleyan, "Breast Cancer Classification Using Moments", in Proc. Signal Processing and Communications Applications Conference (SIU) 20th, Muğla, Turkey, 18-20 April, 2012, pp. 1-4, doi:10.1109/SIU.2012.6204778.
- [45] M. Karabatak, M. C. İnce, E. Avcı, "An Expert Sytem for Diagnosis Breast Cancer Based on Principal Component Analysis Method", in Proc. Signal Processing, Communication and Applications Conference, SIU 2008, IEEE 16th, Aydın, Turkey, 20-22 April, 2008, pp. 1-4, doi:10.1109/SIU.2008.4632642.