



Finding Minimum and Maximum Values of Variables in Mathematical Equations by Applying Firefly and PSO Algorithm

Rouwayda Nadhim Salih¹, Marwa Adeeb Al-jawaherry²

¹ Department of pharmacy, Northern technical university, Technical institute, Kirkuk, Iraq.

² Computer Science Department, College of Computer Science and Mathematics, University of Tikrit, Tikrit, Iraq.

<https://doi.org/10.25130/tjps.v25i5.298>

ARTICLE INFO.

Article history:

-Received: 14 / 1 / 2020

-Accepted: 25 / 7 / 2020

-Available online: / / 2020

Keywords: Optimization problems, Swarm intelligence, Firefly algorithm, Particle Swarm Optimization (PSO) algorithm, Mathematical equations.

Corresponding Author:

Name: Rouwayda Nadhim Salih

E-mail: rouwaydasa@gmail.com

Tel:

ABSTRACT

Computers are often used when problems are big or hard to solve. However, traditional ways to find solutions are not enough when problems are very serious. Hence, turning to nature may be the answer to find solutions for these problems. Artificial intelligence tries to simulate creatures and activities in nature turning their techniques to find solutions for a given problem into an algorithm. Although there are many algorithms have been developed, whose works are inspired by nature, there has been continuous researches aimed at finding better and faster algorithms. Many mathematical optimization problems can be solved by Swarm intelligence algorithms. The aim of this algorithm is to get the optimum solution by repeated searches whose main concern is to discover the area related to solution. In this paper, Firefly Algorithm (FA) and Particle Swarm Optimization (PSO) algorithm are executed. The two algorithms are implemented to find the minimum and maximum values of the mathematical equations. Users of the proposed system are able to read the equation directly through the execution time, by displaying and analyzing the obtained results which show that the FA algorithm has a better performance than PSO algorithm. In addition, the program is executed by MATLAB.

I. Introduction

Optimization is among the important way to solve many scientific and engineering problems. Through the continuous development in recent years, many optimization approaches to solve these problems have been developed; most of them inspire their work from nature [1]. Algorithms that inspire their work from nature are among the strongest algorithms to solve the recent problems. Examples of this kind of algorithms are Particle Swarm Optimization (PSO), whose behaviors depend on the behaviors of swarms such as: shoal of fish and swarms of birds in nature. Recently, this algorithm has been applied to find many solutions of optimization applications [13]. The behavior of natural species is the main inspiration to design and implement many new optimization techniques, such as ants systems which were developed using principle of observation ants in nature. The intelligence of swarm, which is the behavior applied in these algorithms, is a term used to describe the collective behaviors of decentralized and auto organization systems, whether it is natural or

artificial. Therefore, it depends on the interaction among its components inspiring its behaviors from insects. The idea of this algorithm is distributing of a swarm in a limited research area moving randomly to find optimum solution in the area. Generally, when the number of swarms is increasing and the area of research is decreasing, it becomes easier and faster to find the optimum solution. When the number of components of swarm is decreasing and the area of research is increasing, it becomes hard to find the optimum solution [14]. The work of firefly algorithm is inspired by the behaviors of fireflies in nature. Firefly produces short and rhythmical flashes. Flash comes out of bioluminescence represents an activity of each firefly or it may be used as warnings fireflies' works on groups attracting each other by this flash [5][3]. This algorithm is widely used to solve the problems of reliability and repeating [4][16].

II. Problem statement

In this paper, a program that deals with mathematical functions was designed, which cannot be accepted in

advance, where these functions are multivariable. The proposed system calculates the maximum and minimum values of the function variables, as this function is defined and entered directly by the user during the system execution time.

III. Previous works

In 2011, Hanan A. R. Akkar and Firas R. Mahdi proposed adoption of the artificial neural networks using the PSO algorithms as mechanism to optimize the performance of artificial neural networks and to decrease the necessary circuits to build networks. The toolkit of the PSO algorithm was modified in this paper in the Matlab environment to be suitable for the desired application. In the proposed design, the training was carried out out-of-chip and then the design carried the entire training to the chip. In this way, the number of circuits is reduced as little as possible. This paper executes calculation and logic of ALU using the tools of diagramming as an example to apply the digital circuits using artificial neural networks by the PSO algorithm [11].

In 2012, Saibal K. Pal and others applied the firefly algorithm and the PSO algorithm for solving nonlinear mathematical problems. A number of computational experiments were done by the researcher and his colleagues showing obtained results in the form of diagrams explaining the ways of finding the optimum solution. Then, the results of these experiments are analyzed and compared with the best solutions that are obtained so far depending on average of the execution time. The obtained results show that the performance of the firefly algorithm is better than that of the PSO algorithm [14].

In 2014, Amarita Ritthipakdee and et al proposed an improvement of the original firefly algorithm. The proposed algorithm takes into consideration not only the firefly's reaction to light, but also the following factors: the exchange of genes in the firefly, its pheromone, and the wind effect on the dispersion of pheromone. The proposed algorithm has been tested and compared with the original firefly algorithm. It was found that the proposed algorithm was not only more effective but faster than other algorithms [16].

In the year 2019, a common overview of the techniques for dealing with the determinants was presented by Umesh Balade and Deepti Shrimankar. They developed a hybrid algorithm called Stochastic Ranking with Import Firefly Algorithm (SRIFA), applied for solving geometric improvement problems in the real world. This algorithm is applied to enhance 24 functions of 200 CEC and five of common geometric design problem. Through the arithmetical results, it becomes obvious that results obtained from the test of the hybrid algorithms that are already mentioned that (SRIFA) algorithm has better results than that of the basic firefly algorithm [5].

IV. Firefly algorithm (FA)

Firefly algorithm is among the best ways for optimization inspired by nature, specifically by the

behaviors of firefly in the nature. It has been developed by a scholar named Yang in Cambridge University. Each firefly produces a dim light that can be used by fireflies as warning signs. Since 2007, the year in which this algorithm is invented, firefly algorithm has attracted many researches around the world. Recent studies have shown that this algorithm is effective in numerical optimization and combinatorial optimization and it can be better than other algorithms designed in the same field [20].

A. Behavior of fireflies

The light of fireflies in nature is a beautiful scene in summer especially in Tropical and middle zones. There are around two thousands kinds of fireflies and most of them have a light to be short and rhythmical in order not to reach to far distances. Lighting often is unique for each firefly. Flash light is produced by a biological process and the way to produce these flashes is a matter of discussion. Nonetheless, there are two basic functions for these flashes: attraction for breeding and attraction for possible preys. In addition, this flash can be like a cautionary warning. The average of rhythmical light and the amount of lighting time represent a part of a signal system that gather two sexes together. Female responds to the mode produced by males of the same kind. In some kinds, females mimics the mode of flash that is special to attract males in the other kinds of fireflies making males preys for them [3].

B. Function of firefly algorithm

1. Attractiveness

The function of attractiveness for fireflies has its own way that is shown in the decreasing function described in equation (1). Where (r) denotes the distance between any fireflies and the primary value of (r) is zero and the value of attraction is zero too, and γ is an absorption coefficient that controls the reduction of light intensity [15].

$$\beta(r) = \beta_0 e^{-\gamma r^m} \quad \text{with } m \geq 1 \quad (1)$$

2. Distance

If there are two fireflies: firefly (i) and firefly (j), the distance between them will be calculated in the following equation:

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

Where x_i represents the firefly in location (i) within the area of research, and (d) denotes the dimensions of the problem [6] [15].

3. Movement

Equation (3) shows the movement of fireflies which are attracted by other fireflies, eventually having more attraction.

$$X_{i+1} = x_i + \beta_0 e^{(\gamma r)^2} (x_i - x_j)^2 + \alpha (\text{rand} - 0.5) \dots(3)$$

The second term refers to the factor of attraction, where the third term refers to the random distribution, where the coefficient of the random distribution is denoted by (rand) and the value of which is distributed within the scope [0, 1] [9], figure 1 shows firefly algorithm. The following equation represents the primary generation of the fireflies:

$$x_i = LB + rand \cdot (UB - LB) \dots (4)$$

Where (LB) and (UB) denote the minimum and maximum limits for a firefly [7].

1. Algorithm's parameters initialization:
 - Number of fireflies (n_f).
 - β_0, γ, α
 - Maximum number of generations (iterations, n_{itre}).
2. Define the objective function $f(x)$, $x = (x_1, \dots, x_d)^T$.
3. Generate initial population of fireflies x_i ($i = 1, 2, \dots, n$). Light intensity of firefly I_i at x_i is determined by value of objective function $f(x_i)$.
4. While $k < n_{itre}$
5. For $i = 1:n$
6. For $j = 1:i$
7. If ($I_j > I_i$) move firefly i towards firefly j in d -dimension according to Eq. (3); End if.
8. Obtain attractiveness, which varies with distance r according to Eq. (1).
9. Find new solutions and update light intensity
10. End for j .
11. End for i .
12. Rank the fireflies and find the current best
13. End while
14. Find the firefly with the highest light intensity.

Fig. 1: Firefly algorithm [10]

V. Partiale swarm optimization algorithm

PSO algorithm is a combination of swarms or elements in which each element is called a particle that represents a solution. The behavior of the combination is inspired basically by the algorithms that inspire their work from living creature in nature. In addition, this model is inspired from the social behavior of swarms of birds or group of fishes while moving from one place to another. This model was developed by Dr. Eberhart and Dr. Kennedy in 1995 [8].

The purpose of this algorithm is to get the optimum solution and results through simulating the behaviors of birds in their search of best food. Consequently, each system that depends on this algorithm will be represented at the beginning from a random combination of a random solution. Search within this combination includes the optimum solution through the update of generation.

For example, one can imagine a bird looking for food through other birds that refers to a possible feed source, or when a bird remembers a place of food it saw and other near discovery area. It can also be seen as a simulation of a method through which problems can be solved by simulating people we know and we consider successful, and through learning self-dependence. Consequently, our problems will be solved through our experience and others that we know that solved similar problems in similar ways [12].

The PSO algorithm finds quick solutions for nonlinear, invariable and multimedia problems. To explain the work of the algorithm, it is assumed that each single solution is a bird within the distance of search is called particle. All particles have positions and velocities that leads these flying components that fly within the space of problem through following the

best component so far [10]. Swarm represents a possible solution for the problem that can be determined by fitness function. In each iteration, a new location for particle will be evaluated depending on its last position and velocity. Therefore, each particle has one chance to move for each iteration within limited velocity. Therefore, if velocity is too high, a particle will move faster and if the velocity is very slow, a particle will move slowly which makes approaching very slow. At the beginning, the PSO algorithm distributes the particles randomly in the search area, then it uses objective function simply to estimate the suitability of each particles. As a result, each particle has its own location, value of fitness function and velocity. In this case the best value of fitness existed in the swarm identified as the best particle or best individual solution [18] [19]. From what are mentioned earlier it is clear that each particle in the swarm has two basic processes:

1. Velocity update.
2. Position update [2] [17].

Finally, PSO algorithm estimates global best solution, which is the location of particle that gives the minimum and maximum limits from the value of fitness among the entire particle in the group. The following equations explains the concept of the standard PSO algorithm that uses both current global best g_i^* and current individual best x_i^* to get the desired value after many iterations. Figure (2) shows PSO algorithm. In the following section, the variables mentioned in the previous equations will be explained [18]:

$$x_i^{t+1} = x_i^t + v_i^{t+1} \dots (5)$$

$$v_i^{t+1} = wv_i^t + c_1 \varepsilon_1 (x_i^* - x_i^t) + c_2 \varepsilon_2 (g_i^* - x_i^t) \dots (6)$$

$$g_i^* =$$

$gbest[]$, Represents best position within the whole swarm

x_i^* = w = inertia weight
 $pbest[]$, Represents best position of the current component at t , Represents random number
 $x_i^t = t$, Represents position of i in t moment c_1, c_2 , Represent coefficient of acceleration,
 $v_i^t = t$, Represents velocity of i in t moment most of times their values are equal 2

Algorithm 2 Particle Swarm algorithm

```

1: procedure PSO
2:   repeat
3:     for  $i = 1$  to number of individuals do
4:       if  $G(\bar{x}_i) > G(\bar{p}_i)$  then ▷  $G()$  evaluates goodness
5:         for  $d = 1$  to dimensions do
6:            $p_{id} = x_{id}$  ▷  $p_{id}$  is the best state found so far
7:         end for
8:       end if

9:        $g = i$  ▷ arbitrary
10:      for  $j =$  indexes of neighbors do
11:        if  $G(\bar{p}_j) > G(\bar{p}_g)$  then
12:           $g = j$  ▷  $g$  is the index of the best performer in the neighborhood
13:        end if
14:      end for

15:      for  $d = 1$  to number of dimensions do
16:         $v_{id}(t) = f(x_{id}(t - 1), v_{id}(t - 1), p_{id}, p_{gd})$  ▷ update velocity
17:         $v_{id} \in (-V_{max}, +V_{max})$ 
18:         $x_{id}(t) = f(v_{id}(t), x_{id}(t - 1))$  ▷ update position
19:      end for
20:    end for
21:  until stopping criteria
22: end procedure
    
```

Fig. 2: PSO algorithm [18]

VI. The proposed system

A. Description

The firefly algorithm and the PSO algorithm are used to solve many optimization problems. To do so, it is needed to identify the objective function, naming it the firefly algorithm. On the other hand, PSO algorithm is named in fitness. In addition, many needed variables are identified called variables of control.

The objective function will be identified by a user from the interactive interface and the following format includes some variables that can be existed in the objective function:

$$max \rightarrow f(x, y) = x + cy^2 - xy, (x, y) \in (-w, w) \quad (7)$$

w : is the maximum and minimum limits for the firefly (i)

x, y : the value of which is random, while C : refers to a constant number.

In this paper, the firefly algorithm and the PSO algorithm are executed on MATLAB in windows 8 or windows 10, the result will be displayed on the interactive interface. When any technique of metaheuristic is executed, variables of control should be prepared. This is the same for firefly algorithm where it is very important to select the variables of control to find the optimum solution. Our choices for these variables depend on the experimental results.

1. Choosing control variables for firefly algorithm

The variables of control for firefly algorithm are explained in the following way:

1. n_f : is the number of fireflies in all examples since its value = 45 which is tested because the increasing of which to more than 100 will not change the result notably and the execution time will increase with no improvement in results.
2. n_{iter} : is the number of iteration, which is another variable of control for firefly algorithm that should be identified the algorithm to be implemented until the solution that is near to the optimum one will be found in low mistakes. In all examples $50 = (n_{iter})$, is identified and it is found that the result is suitable. When the value of n_{iter} is increased to more than 50, this will not change the result.
3. β_0 : is the primary value for (attractiveness). In many proposed optimization problems ($0.1 = \beta_0$). In this paper, bigger value is used, which yields better result.
4. γ : is absorption coefficient whose value in this research is $\gamma = 1$, this value increases the velocity of algorithm the solution to find the optimum solution.
5. μ : is the value of possible coefficient, which can be an positive number. Its value in the search is $\mu = 0.1$
6. α : is the value of random variable in which the variable control has a value of any number between ($w, -w$). The rate of randomness is specified with a value of α and the variable, which is very important since it allows to have a new solution in order not to

add a low local limit for it. In this research, the value of $\alpha = 0.1$, this value is selected to avoid the disturbance of fireflies.

At the beginning, values for the variables of control will be selected. Then, the algorithm is executed until it reaches iter=50, to remove the random effects and avoids the early approaching. 20 independent operations are executed. After that, the firefly with a best value of fitness is selected as an optimum solution for the given problem.

2. Choosing control variables for pso algorithm

The variables of control for the PSO algorithm are explained as follows:

The PSO algorithm is prepared with a group of random components (solutions), and then the best solution through the update of generation is looked for.

1-N: is the number of particles in all examples whose values = 45.

2-IterMax: is the number of iteration, which is another variable of control for the PSO algorithm. In all examples, the number of iteration is identified iter =50 (in this algorithm, when the number of iterations is increasing, the result will be best and more accurate). Within each iteration, each component is being updated ith the group through following values:

“*pbest*”: represents the best suitable value that (i) reaches so far, “*gbest*”: represents the best suitable value within the whole swarm, the best component within the whole swarm.

“*lbest*”: when a component in the swarm is taking position within a group, as for other components, the best value within this partial groups represents the best local value called lbest. After finding the best values, components are modified in terms of their velocity, position according to equations a and b.

$$v[] = c3 * v[] + c1 * rand() * (pbest[] - present[]) + c2 * rand() * (gbest[] - present[]) \dots (a)$$

$$present[] = present[] + v[] \dots (b)$$

Where v[]: represents the velocity of a particle.

present[]: represents the position of (i) in k moment.

Pbest[]: represents the best position for the current component of the swarm.

Gbest[]: represents the best position within the whole swarm.

Tand: represents a random number within the scope [0, 1]

c1, c2, c3: acceleration coefficients, they often have the values of c1+c2+c3=1.

The values of variables will be identified, then, the algorithm is executed until it reaches iter =50. After that, a particle with the best fitness value will be selected as an optimum solution for the problem.

B. System interfsces

The following figures explain the whole cases existed in the system:

1. First case: shows the main interface containing the whole parts existed in the system.

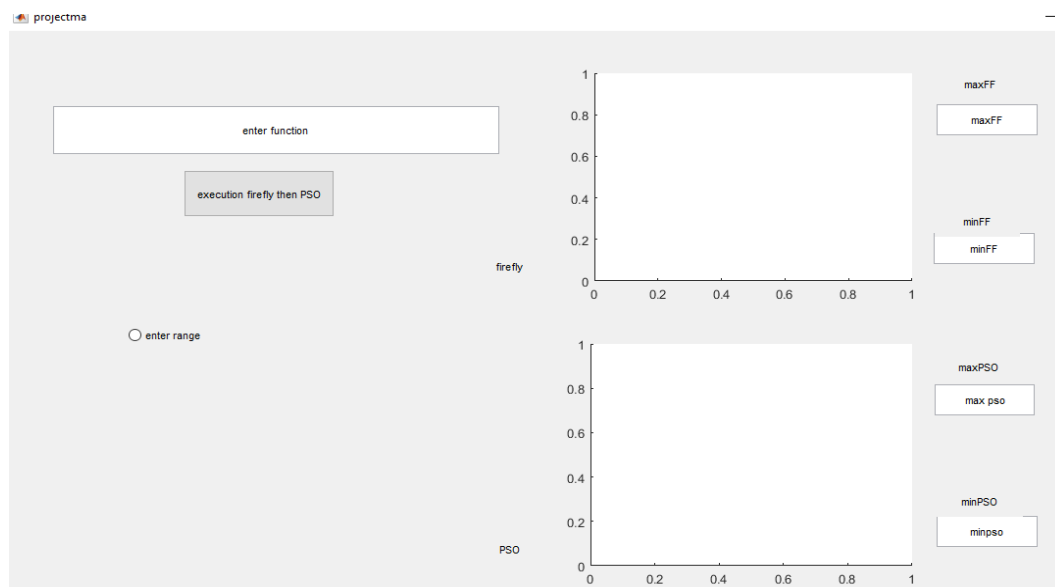


Fig. 3-a: The main interface for the system without entering scope value

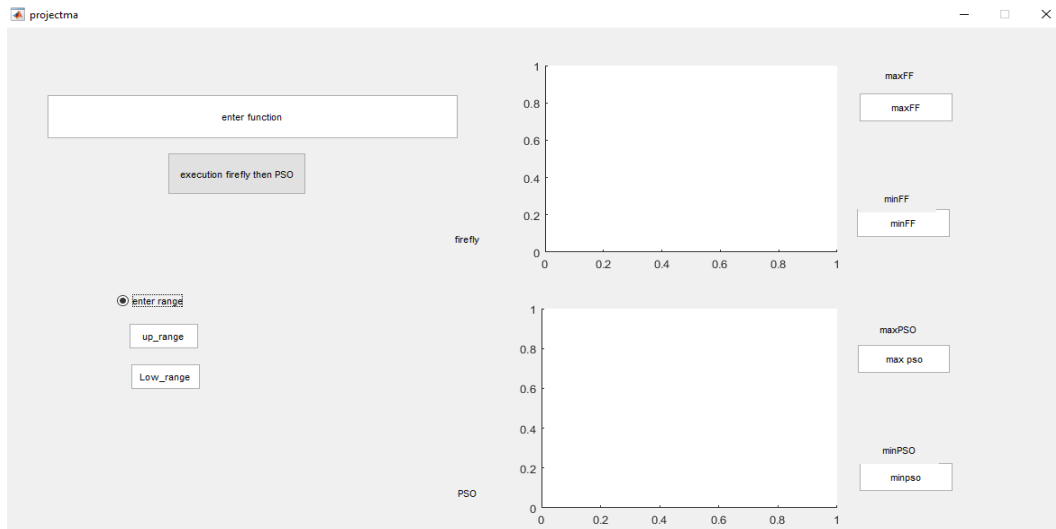


Figure 3-b: The main interface for the system with scope values being entered

2. Second case: entering the equation directly by user.

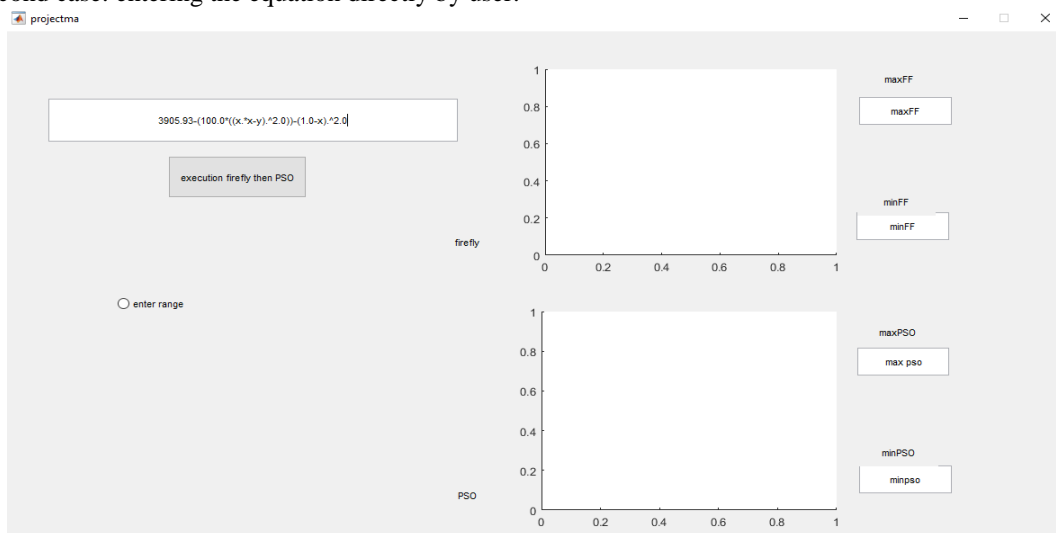


Fig. 4: Entering equations through user interface

3. Third case: entering scope value.

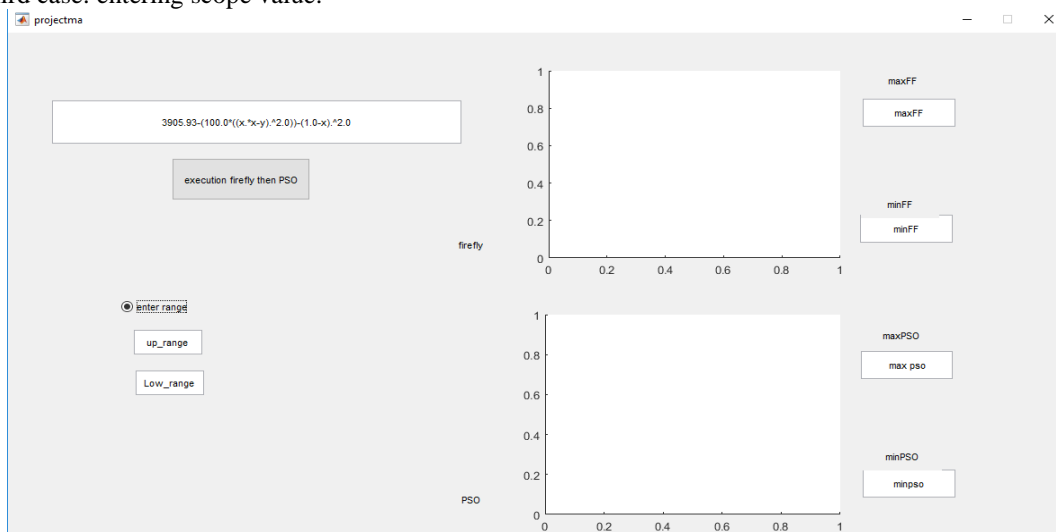


Fig. 5-a: Entering scope value.

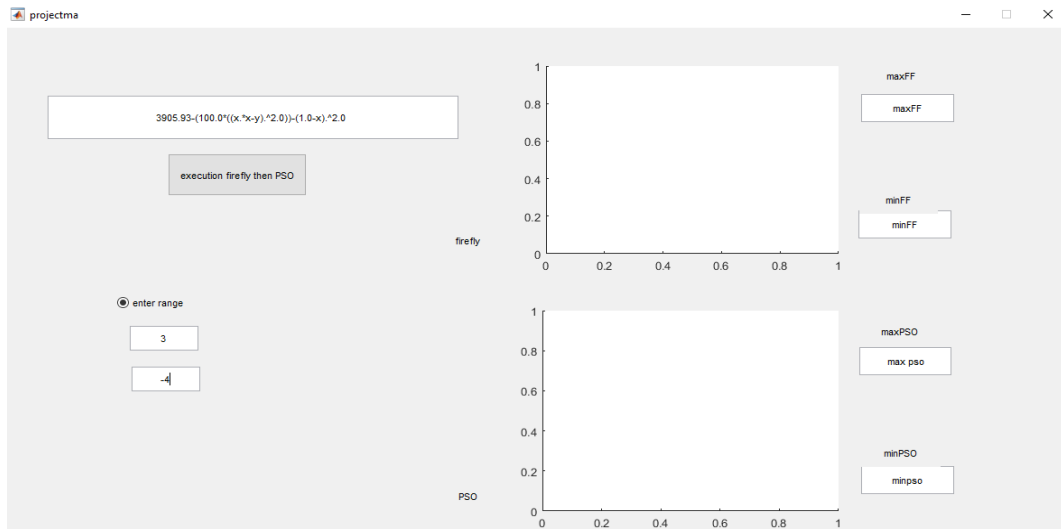


Fig. 5-b: Entering scope value.

4. Fourth case: Executing Firefly and PSO algorithms.

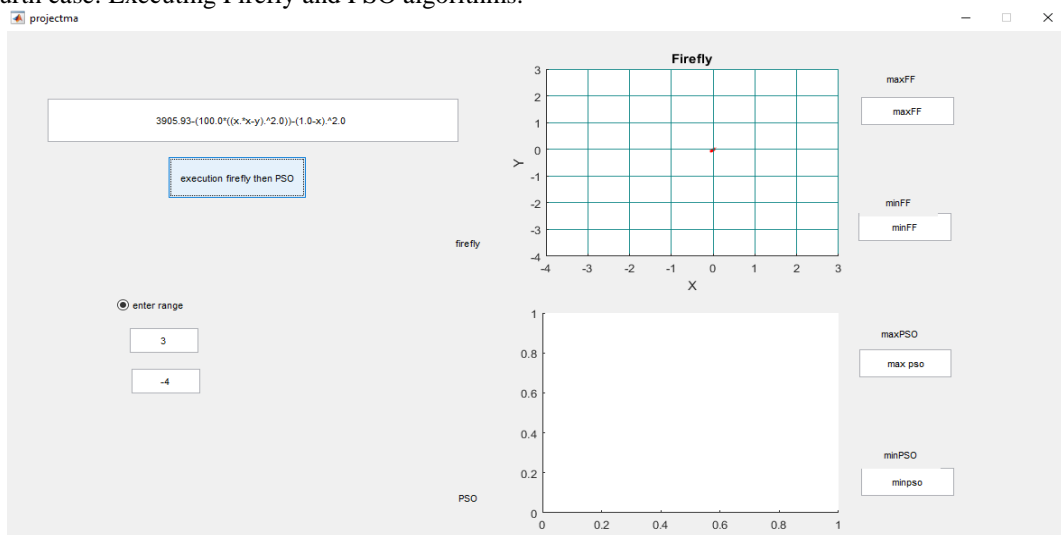


Fig. 6-a: Execution of the Firefly algorithm

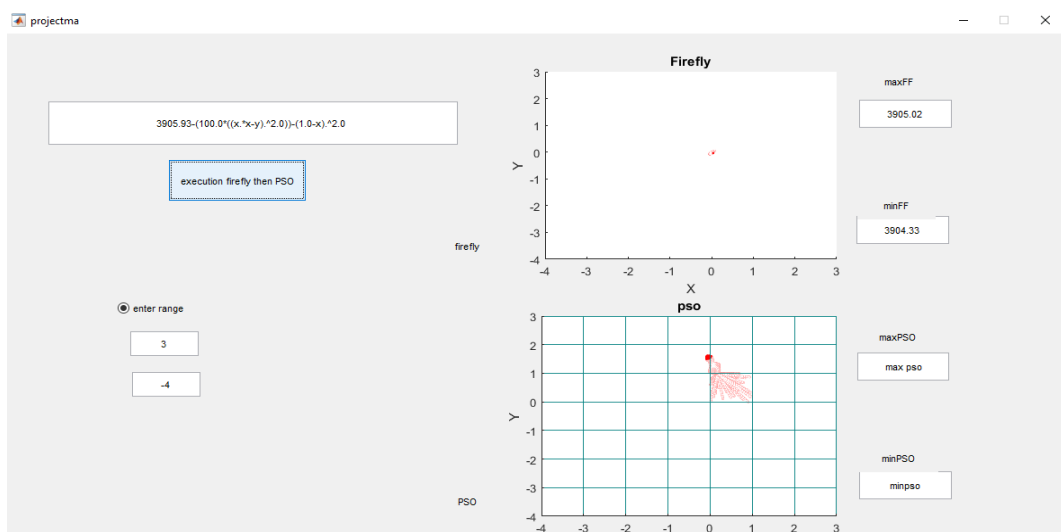


Fig. 6-b: Execution of the Firefly algorithm

5. fifth case: The case of entering a wrong equation.

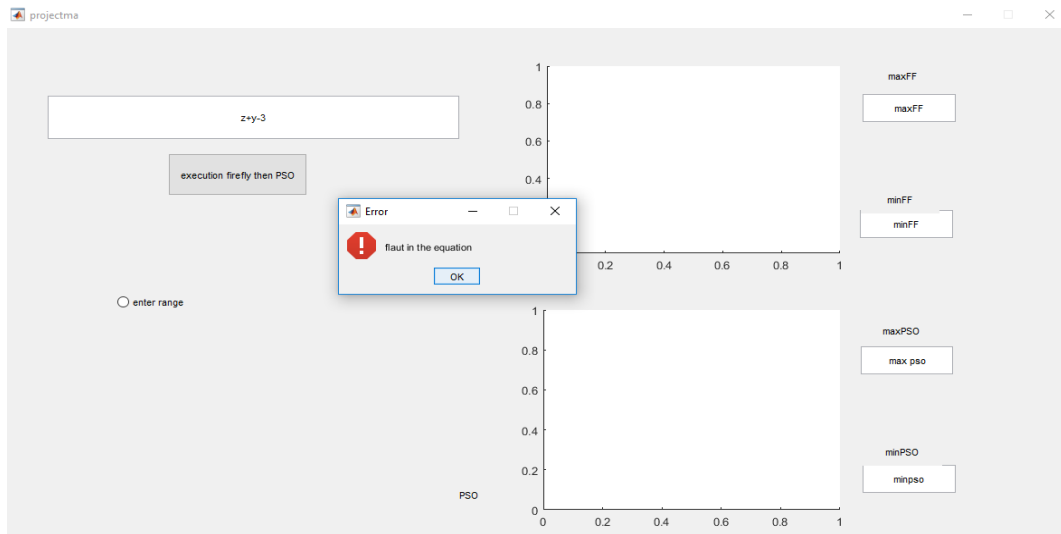


Fig. 7: The case of entering a wrong equation

6. Sixth case: execution of the first equation

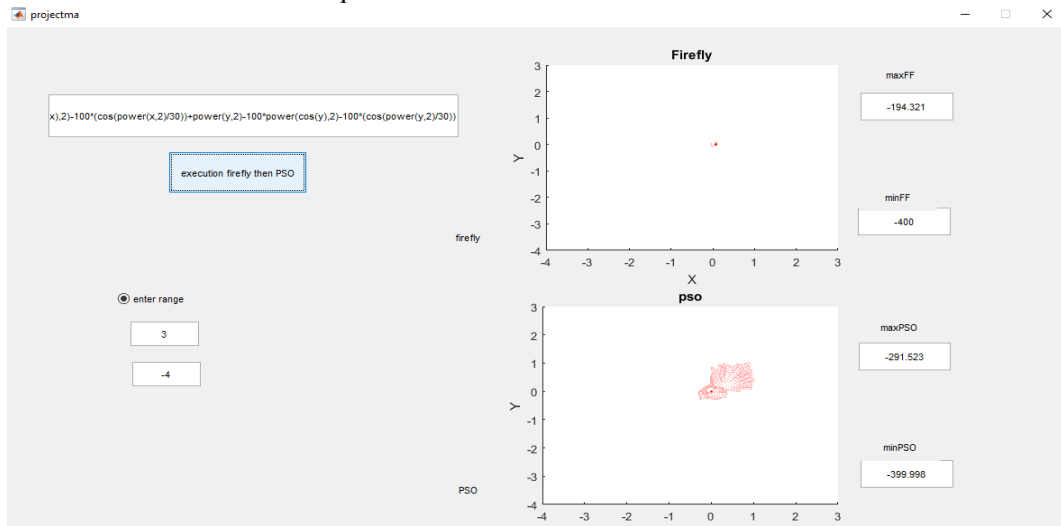


Fig. 8: Execution of the first equation

7. Seventh case: Execution of second equation.

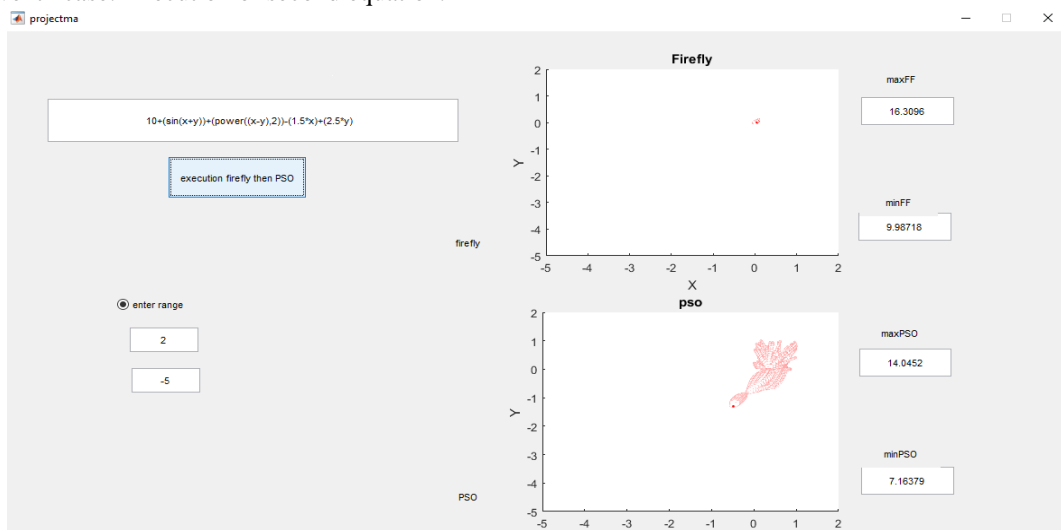


Fig. 9: Execution of second equation.

8. Eighth case: Execution of the third equation.



Fig.10: Execution of the third equation

9. Ninth case: execution of a equation without entering the scope value by a user and the system wil

I take the scope value from [10,10-] automatically.



Fig. 11: the case of the scope is not entered by the user.

VII. Conclusion

By comparing the two algorithms, it has become clear that the firefly algorithm is better to find the optimum solution. When equations are executed, the results of the firefly algorithm are better than that of the PSO algorithm. The experiments done on the two algorithms have shown that when the number of iteration is increasing, the firefly algorithm helps to find the optimum solution faster, and then it is stopped at certain iteration. For the PSO algorithm, increasing the number of iteration leads to improve

the result, this means that the firefly algorithm is faster to find the optimum solution than that of the PSO algorithm.

It is possible to propose many future works in this paper such as adding other algorithms, like bee swarm, increasing functions related to comparison, making computations to compare between the algorithms and improving the interfaces. In addition, the FA algorithm can be executed using hardware devices that will increase time and decrease energy used to run the algorithm.

References

- [1] Abdullah, A.; Deris, S.; Mohamad, M.S. and Mohamad, S.Z.M. (2012). A New Hybrid Firefly Algorithm for Complex and Nonlinear Problem, *The 9th International Conference for Symposium on Distributed Computing and Artificial Intelligence (DCAI)*, March 2012: p 673–680.
- [2] Al Hwaitat A.K.; Almaiah M.A.; Almomani O.; Al-Zahrani M.; Al-Sayed R.M.; Asaifi R. M.; Adhim K.K.; Althunibat A. and Alsaaidah A. (2020). Improved Security Particle Swarm Optimization (PSO) Algorithm to Detect Radio Jamming Attacks in Mobile Networks. *International Journal of Advanced Computer Science and Applications (IJACSA)*, **11(4)**:614-625.
- [3] Alomoush W.; Omar K.; Alrosan A.; Alomari Y.M; Albashish D. and Almomani A. (2020). Firefly photinus search algorithm, *Journal of King Saud University Computer and Information Sciences*, **32 (2020)**:599–607.
- [4] Asokan K. and Ashok K.R. (2014). Application of Firefly algorithm for solving Strategic bidding to maximize the Profit of IPPs in Electricity Market with Risk constraints. *International Journal of Current Engineering and Technology*, **4(1)**: 37-44.
- [5] Balande U. and Shrimankar D. (2019). SRIFA: Stochastic Ranking with Improved Firefly Algorithm for Constrained Optimization Engineering Design Problems. *Mathematics Journal (MDPI)*, **7(250)**:1-26.
- [6] Benabid R.; Zellagui M., Chaghi A. and Boudour M. (2014). Application of Firefly Algorithm for Optimal Directional Over current Relays Coordination in the Presence of IFCL, *International Journal of Intelligent Systems and Applications*, **7(1)**:44-53.
- [7] Dhillon M. and Goyal S. (2013). PAPR Reduction in Multicarrier Modulations Using Firefly Algorithm. *International Journal of Innovative Research in Computer and Communication Engineering*, **1(5)**: 1272.
- [8] Eberhart R., and Kennedy J. (1995). A new optimizer using particle swarm theory. *The 6th International Conference for Symposium on Micromachine and Human Science*, Publisher: IEEE, Nagoya, Japan: p 39-43.
- [9] Farook S. and Raju P. S. (2014). Metaheuristic Algorithms for Capacitor Siting and Sizing to Improve Voltage Profile. *International Electrical Engineering Journal (IEEJ)*, **5(1)**:1208-1215.
- [10] Frederic N. and Yang Y. (2017). Analysis of Particle Swarm Optimization and Genetic Algorithm based on Task Scheduling in Cloud Computing Environment. *International Journal of Advanced Computer Science and Applications (IJACSA)*, **8(1)**:19-25.
- [11] Hanan A.R.A. and Firas R. M. (2011). Training Artificial Neural Networks by PSO to Perform Digital Circuits Using Xilinx FPGA. *Eng. & Tech. Journal*, **29(7)**:1329- 1344.
- [12] Jancauskas V. (2014). Empirical Study of Particle Swarm Optimization Mutation Operators. *Journal of Baltic J. Modern Computing*, **2(4)**:199–214.
- [13] Jiang S.; Zhang C. and Chen S. (2020). Sequential Hybrid Particle Swarm Optimization and Gravitational Search Algorithm with Dependent Random Coefficients. *Journal of Mathematical Problems in Engineering*, **2020 (Article ID 195)**:1-17.
- [14] Kwiecien J. and Filipowicz B. (2012). Firefly algorithm in optimization of queuing systems. *Bulletin of the Polish Academy of Sciences, Technical Sciences*, **60(2)**: 363-368.
- [15] Mahmood R.Z. and Al-jawaherry M.A. (2017). Firefly Algorithm Implementation Based on Arduino Microcontroller. *International Journal of Computer Science and Information Security (IJCSIS)*, **15(12)**:283-288.
- [16] Ritthipakdee A.; Thammano A. Premasathian N. and Uyyanonvara B. (2014). An Improved Firefly Algorithm for Optimization Problems. *ADCONP, Hiroshima*, **2014(2)**:159-164.
- [17] Rostami A. and Lashkari M. and Branch F. (2014). Extended PSO algorithm for improvement problems K-Means clustering algorithm. *International Journal of Managing Information Technology*, **6(3)**:17-29.
- [18] Saibal K. Pal; Rai C.S and Singh A.P. (2012). Comparative Study of Firefly Algorithm and Particle Swarm Optimization for Noisy Non Linear Optimization Problems. *I.J. Intelligent Systems and Applications*, **10(6)**:50-57.
- [19] Zhan, S. and Huo H. (2012). Improved PSO-based task scheduling algorithm in cloud computing. *Journal of Information and Computational*, **9(13)**: 3821–3829.
- [20] Zhui M.Y.; Bin M.Y. and yan Z.Q. (2013). Optimal Choice of Parameters for Firefly Algorithm. *Fourth International Conference on Digital Manufacturing & Automation*, Qingdao, China, 29-30 June 2013: p. 887-892.

ايجاد الحد الأدنى والأقصى لقيم المتغيرات في المعادلات الرياضية من خلال تطبيق خوارزميتي البراعة و سرب الطيور

رويدة ناظم صالح احمد¹ ، مروى اديب محمد الجواهري²

¹قسم الصيدلة ، المعهد التقني ، الجامعة التقنية الشمالية ، كركوك ، العراق

²قسم الحاسوب ، كلية علوم الحاسوب والرياضيات ، جامعة تكريت ، تكريت ، العراق

الملخص

تستخدم أجهزة الكمبيوتر غالباً عندما تكون المشكلات كبيرة أو يصعب حلها. ومع ذلك، فإن الطرق التقليدية لإيجاد الحلول ليست كافية عندما تكون المشاكل جديّة للغاية. وبالتالي، قد يكون اللجوء إلى الخوارزميات المستوحاة من الطبيعة هو الخيار الأفضل لإيجاد حلول لهذه المشكلات. يحاول الذكاء الاصطناعي محاكاة سلوك المخلوقات والأنشطة في الطبيعة وتحويل تقنياتهم إلى خوارزمية لإيجاد حلول لمشكلة معينة. على الرغم من أن هناك العديد من الخوارزميات التي تم تطويرها، والتي تستلهم أعمالها من الطبيعة، فقد كانت هناك أبحاث مستمرة تهدف إلى إيجاد خوارزميات أفضل وأسرع. يمكن حل العديد من مشكلات التحسين الرياضي عن طريق خوارزميات الذكاء والسرب ، والتي استوحى عملها من سلوكيات المخلوقات في الطبيعة.

الهدف من هذه الخوارزميات هو الحصول على الحل الأمثل من خلال عمليات البحث المتكررة التي تتمثل اهتماماتها الرئيسية في اكتشاف المنطقة ذات الصلة بالحل. في هذا البحث ، تم تنفيذ خوارزمية البراعة (FA) وخوارزمية سرب الطيور (PSO) ، إذ إن خوارزمية البراعة هي أكثر الخوارزميات شيوعاً ، وعملها مستوحى من سلوك اليراعات في الطبيعة. من ناحية أخرى ، تعمل خوارزمية سرب الطيور على شكل مجاميع ، كما أن عملها مستوحى من السلوكيات الاجتماعية للطيور أو الأسماك.

تم تطبيق الخوارزميتين للعثور على الحد الأدنى والحد الأقصى لقيم المعادلات الرياضية. يمكن لمستخدمي النظام المقترح قراءة أي معادلة رياضية مباشرة خلال وقت التنفيذ ، تم عرض وتحليل النتائج التي تم الحصول عليها والتي توضح أن خوارزمية FA لديها أداء أفضل من خوارزمية PSO بالإضافة إلى ذلك ، تم تطوير البرنامج باستخدام لغة MATLAB.

الكلمات المفتاحية : مسائل التحسين ، ذكاء سرب ، خوارزمية البراعة ، خوارزمية سرب الطيور ، المعادلات الرياضية.