Journal of Operation and Automation in Power Engineering

Vol. 8, No. 1, Feb. 2020, Pages: 57-64

http://joape.uma.ac.ir

FOA: 'Following' Optimization Algorithm for Solving Power Engineering Optimization Problems

M. Dehghani[1,*], *M. Mardaneh*[1], *O. P. Malik*[2]

[1]*Department of Electrical and Electronics Engineering, Shiraz University of Technology, Shiraz, Iran.*
[2] *Department of Electrical Engineering, University of Calgary, Calgary Alberta Canada*

Abstract- *These days randomized-based population optimization algorithms are in wide use in different branches of science such as bioinformatics, chemical physics and power engineering. An important group of these algorithms is inspired by physical processes or entities' behavior. A new approach of applying optimization-based social relationships among the members of a community is investigated in this paper. In the proposed algorithm, search factors are indeed members of the community who try to improve the community by 'following' each other. FOA implemented on 23 well-known benchmark test functions. It is compared with eight optimization algorithms. The paper also considers for solving optimal placement of Distributed Generation (DG). The obtained results show that FOA is able to provide better results as compared to the other well-known optimization algorithms.*

Keyword: *heuristic algorithms, following, following optimization, social relationships, optimization.*

1. INTRODUCTION

The outgrowing magnitude of optimization problems and the urge to find a response as quickly as possible has resulted in the diminishing use of classical methods in solving optimization problems. Currently, use of randomized search algorithms, instead of searching all dimensions of a problem, is becoming more common. In this regard, application of Heuristic Search Algorithms (HSA) has noticeably increased lately[1-4]. Heuristic algorithms have proven their ability in many fields of science, like transportation [5], bioinformatics [6], energy [7], chemical physics [8], electronics [9], and other related fields. Finding a mathematical model to be applied in HSA is difficult and even impossible. Therefore, these algorithms are called the black box optimization algorithms [10].

In population-based methods, members interact and transfer information by applying different methods, like Genetic Algorithm (GA) which is inspired by genetics and evolution (1975) [11], Artificial Immune System (1986) by imitating the human body immune system [12], Ant Colony Optimization (ACO) by simulating ant behavior in searching food (1991) [13], and Particle Swarm Optimization (PSO) by imitating birds' social behavior (1995) [14].

In the present paper, a new approach of applying the mutual relationships between the community members is used in designing the optimization algorithm. In this algorithm, the 'following' aspect of the community members is used to design an optimization algorithm named as the Following Optimization Algorithm (FOA). A brief history of the heuristic optimization methods is presented in section two. The proposed algorithm is described in section three and section four encompasses the related explanations. Results are described in section five and conclusions are presented in section six.

2. A BRIEF HISTORY OF META-HEURISTIC PTIMIZATION ALGORITHMS

A meta-heuristic optimization algorithm is a key to finding a proper answer to the optimization problem that can solve the problem by having the minimum minute information in the most immediate possible time. In ancient Greece the word "heuristic" meant "to know", "to find", "to discover", or " to guide an investigation" [15, 16]. In a more complete definition, a heuristic method is a strategy that neglects a part of information to arrive at a quick response with the highest precision by the maximum time saving in comparison to the complex methods [17].

Heuristic search algorithms are those algorithms that are inspired and formed by biological and physiological processes of mother nature. Most of these algorithms

Received: 03 Dec. 2018
Revised: 02 Apr. 2019
Accepted: 28 May. 2019
*Corresponding author: Mohammad Dehghani
E-mail: adanbax@gmail.com (M. Dehghani)
Digital object identifier: 10.22098/joape.2019.5522.1414
Research Paper
© 2020 *University of Mohaghegh Ardabili. All rights reserved.*

function within the population. In the recent past, a large number of population-based algorithms, derived from social interactions among community members, have been presented. The most famous and widely-used are: Genetic Algorithm (GA) [11], Simulated Annealing (SA) [18], Harmony Search (HS) [19], Artificial Immune System (AIS) [20], Ant Colony Optimization (ACO) [21], Particle Swarm Optimization (PSO) [14], and Bacterial Foraging Algorithm (BFA) [22]. GA is modelled by genetic laws based on the Darwin's theory [11]. SA is proposed based on the annealing processes in metallurgy [18]. HS is an algorithm in which the process of improving melody is taken into account by the composer while composing [19]. AIS is designed by imitating the human body immune system [20]. ACO simulates ants' behaviors in searching food [21]. PSO depicts the social behavior of a flock of birds while migrating [14]. This algorithm simulates the co-operation among birds' communities. Each particle tries to arrive at the optimum position within the search space by using its own previous experience and by taking the best advantage of consultations received from its neighboring particles [23]. BFO is inspired by observing the social behavior of I-coil bacteria while searching for food [22]. All mentioned algorithms use the statistical property and the randomized phenomena in their application as exists in nature. In some Central Force Optimization Algorithms, known as metaphors of the universal law of gravity, the mentioned randomized phenomena are not used. This means that these algorithms have the deterministic property [24].

Despite classical methods, the heuristic search methods function randomly and they search the searching space in a parallel mode. Besides, they do not use the space gradient information while classical methods do. Indeed, heuristic search-based methods merely use the suitability function to navigate their search. However, they can arrive at the answer of the optimization problem since their intelligence is known as the swarm intelligence. The swarm intelligence appears in cases where there is a population of unfitted elements each of which shows a simple behavior under certain circumstances and has mutual influences on each other locally. The members' local interactions make unexpected para-local effects. Thus, the whole system can arrive at the solution without having a central controller.

It is proper to note that the members' behavior can organize the system internally by making use of properties like the positive feedback, the negative feedback, a balance between exploration and exploitation, and some other interactions. This internal organization is called self-organization [25]. Although many heuristic algorithms have been introduced, improved and used in various scientific fields by investigators, there still is no algorithm to satisfactorily respond and optimize all problems of both engineering and other scientific fields. This article is an attempt to find a new heuristic algorithm by applying which problems of the previous algorithms can be resolved. The proposed method and its details are described in the following section.

3. FOLLOWING OPTIMIZATION ALGORITHM

In the approach proposed here, the optimization process is done by the aid of the chosen community social relationships within an artificial system during a discrete time. The system space is the same as the domain of problem definition. Hence, the community members are used as tools to convey information. Besides, the designed optimum finder can be used to solve any optimization problem in which a response can be defined as a position in space and its similarity with other responses can be declared as a comparison among members regarding their social standing. Noticeably, members' social standing is determined concerning the objective function. FOA is defined in two general steps: 1- making an artificial system with the discrete time within the problem space, the initial navigation of members, determining the governing laws and principles, arranging parameters, 2- passing time to arrive at the stop time.

3.1. Making system, determining principles and organizing parameters

In the first step, the system space is determined. This space includes a multi-dimensional coordinate within the problem definition space. Each point of this space is a response to the problem. Searching elements are a set of members who live in the community. Indeed, each member interacts with all other members. Besides, each member has a social standing characteristic that is a position in space which indeed is the answer to the optimization problem.

After making the system, its governing principles are determined. In the present paper, it is supposed that the community members follow each other. Now consider the system as a set of m members. The position of any member is a point in the space where it is an answer to the optimization problem. In relation 1, the d dimension of member i is shown as x_i^d.

$$X_i = (x_i^1, \dots, x_i^d, \dots, x_i^n) \quad (1)$$

At first, the initial position of the community members is determined randomly within the problem definition space. This community paves the way to the balance state (response). concerning interactions held among members. Now:

$$x_i^d = (1-fr)x_{i,0}^d + fr\, x_{leader}^d \quad (2)$$

$$f = 1 - exp\,(\frac{-t^{1.5}}{T}) \quad (3)$$

$$x_{leader} = \begin{cases} minimization: location\ of\ \min(fit) \\ maximization: location\ of\ \max(fit) \end{cases} \quad (4)$$

In the above equations, $x_{i,0}^d$ refers to the initial balance point along with the d dimension of member i, r represents a random number with a uniform distribution within $[0-1]$ span used to preserve the search random state. x_{leader} is the highest member of the society regarding its social standing who leads the community and indeed all members follow him/her. Moreover, x_{leader}^d is the *d*-th dimension of the position of the mentioned leader. Symbol '*f*' represents the 'following' co-efficient, *t* is the iteration count, *T* is the maximum number of iterations and *fitness* is the community fitness vector.

3.2. Passing time and the parameters updating process

At the beginning of making the system, each member is randomly placed in any point across (throughout) the space where it is considered as the answer to the problem. At any moment of time, each member is assessed and his/her displacement is computed by operating relations (1) through (4). In the next iteration, the given member is placed in the computed position. The system parameter is the following co-efficient f that is updated in each phase based on relation (3). The stop position can be determined after passing a finite time. Different steps of the following optimization algorithm are shown below and the flowchart is shown in Fig. 1.

1- Determine the system space and set initial parameters.
2- Generate initial population.
3- Community assessment.
4- Determine the community leader.
5- Update parameter f.
6- Update members' position.
7- Repeat stages 3 to 6 until the stopping criterion is satisfied.
8- The ending phase.

4. CHARACTERISTICS OF THE FOA

In the proposed algorithm, a new method of optimization is devised concerning the community members' obedience and behaviour in following each other. In this algorithm, a set of people search the chosen space randomly. Social standings are used as tools to convey information. Each member affected by the community can approximately understand his/her surrounding space. The algorithm must be developed such that the members' positions are improved as time passes. The applied strategy to fulfil this aim is to organize the following co-efficient '*f*'. Among the proposed solutions to optimize an algorithm, improving its exploring power has the highest importance.

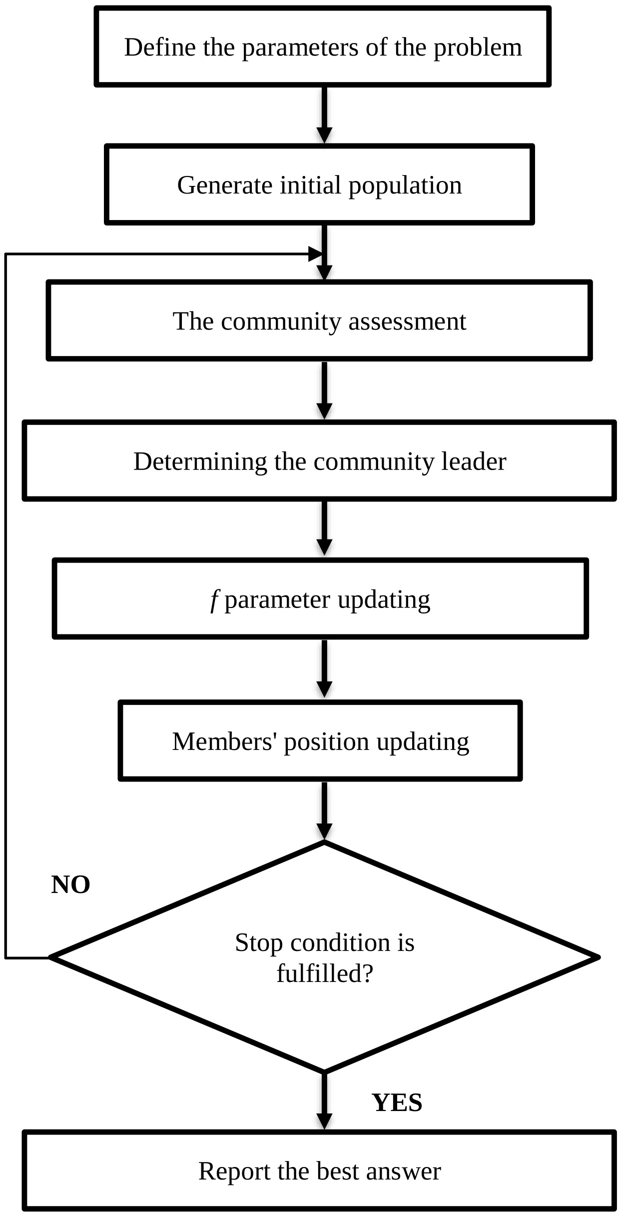

Fig. 1. 'Following' optimization algorithm flowchart.

In optimization two challenges exist: exploration and exploitation. In the realm of exploration, each optimization algorithm must have enough power to extensively search the search space and its search should not be limited to some restricted locations [25].

On the other hand in the realm of exploitation, the algorithm's ability in exploring/discovering the optimum points is under focused. In population-based algorithms, there is a need to comprehensively search space at the very early stages of running the algorithm. Besides, the algorithm must stress better search during initial iterations. However, as time passes, the algorithm's exploration/detection ability becomes paramount. Thus, the algorithm must find the position of each point by the aid of the population findings [26]. Interestingly, the above algorithm has the power to search space by considering a suitable number of members. Here, the proposed solution to improve and fortify the algorithm detection speed is indeed the following co-efficient effect. To fulfil this aim, the following co-efficient *f* is controlled by running relation (3). During the early iterations of the algorithm, there is a severe need for a proper search though it can arrive at better responses as time passes. Thus, the value of the following co-efficient *f* is controlled as a time variable entity. Accordingly, a proper value is chosen for the following co-efficient *f* at the beginning of running the algorithm that is increased as time passes to arrive at its maximum value.

Based on the value of the following co-efficient *f* in each iteration, the community members follow the member who has the highest social standing. During initial iterations, the following co-efficient is set to its minimum rate that makes the search space well explored in order to prevent the algorithm from settling in the local optimum positions. The following co-efficient *f* grows bigger as time passes. Since it is known that people gather around better positions with the passing of time and since it is necessary to explore space with smaller and more precise steps, members' impressibility and influence on each other is increased as the time goes. Thus, it is expected to view members at better positions as the time passes.

5. RESULTS

5.1. Benchmark test functions

Performance of the proposed algorithm is assessed by applying 23 standard criterion functions [27]. The standard benchmark test functions are shown in Tables 1 through 3.

Table 1. Unimodal test functions.

$F_1(x)=\sum_{i=1}^{m} x_i^2$	$[-100,100]^m$
$F_2(x)=\sum_{i=1}^{m} \lvert x_i\rvert + \prod_{i=1}^{m} \lvert x_i\rvert$	$[-10,10]^m$
$F_3(x)=\sum_{i=1}^{m}\left(\sum_{j=1}^{i} x_i\right)^2$	$[-100,100]^m$
$F_4(x)=\max\{\lvert x_i\rvert, \quad 1\le i\le m\}$	$[-100,100]^m$
$F_5(x)=\sum_{i=1}^{m-1}[100(x_{i+1}-x_i^2)^2+(x_i-1)^2)]$	$[-30,30]^m$
$F_6(x)=\sum_{i=1}^{m}([x_i+0.5])^2$	$[-100,100]^m$
$F_7(x)=\sum_{i=1}^{m} ix_i^4 + random(0,1)$	$[-1.28,1.28]^m$

Table 2. Multimodal test functions.

$F_8(x)=\sum_{i=1}^{m} -x_i \sin(\sqrt{\lvert x_i\rvert})$	$[-500,500]^m$
$F_9(x)=\sum_{i=1}^{m}[\,x_i^2-10\cos(2\pi x_i)+10]$	$[-5.12,5.12]^m$
$F_{10}(x)=-20\exp\left(-0.2\sqrt{\frac{1}{m}\sum_{i=1}^{m} x_i^2}\right)-\exp\left(\frac{1}{m}\sum_{i=1}^{m} cos(2\pi x_i)\right)+20+e$	$[-32,32]^m$
$F_{11}(x)=\frac{1}{4000}\sum_{i=1}^{m} x_i^2-\prod_{i=1}^{m} cos\left(\frac{x_i}{\sqrt{i}}\right)+1$	$[-600,600]^m$
$F_{12}(x)=\frac{\pi}{m}\left\{10\sin(\pi y_1)+\sum_{i=1}^{m}(y_i-1)^2[1+10\sin^2(\pi y_{i+1})]+(y_n-1)^2\right\}+\sum_{i=1}^{m} u(x_i,10,100,4)$ $u(x_i,a,i,n)=\begin{cases} k(x_i-a)^n & x_i>-a \\ 0 & -a<x_i<a \\ k(-x_i-a)^n & x_i<-a \end{cases}$	$[-50,50]^m$
$F_{13}(x)=0.1\left\{\sin^2(3\pi x_1)+\sum_{i=1}^{m}(x_i-1)^2[1+\sin^2(3\pi x_i+1)]+(x_n-1)^2\,[1+\sin^2(2\pi x_m)]\right\}$ $+\sum_{i=1}^{m} u(x_i,5,100,4)$	$[-50,50]^m$

Table 3. Multimodal test functions with fixed dimension.

$F_{14}(x) = \left(\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^{2}(x_i - a_{ij})^6}\right)^{-1}$	$[-65.53,65.53]^2$
$F_{15}(x) = \sum_{i=1}^{11} \left[a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4}\right]^2$	$[-5,5]^4$
$F_{16}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	$[-5,5]^2$
$F_{17}(x) = \left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)cosx_1 + 10$	$[-5,10] \times [0,15]$
$F_{18}(x) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \times [30 + (2x_1 - 3x_2)^2 \times (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	$[-5,5]^2$
$F_{19}(x) = -\sum_{i=1}^{4} c_i \exp(-\sum_{j=1}^{3} a_{ij}(x_j - P_{ij})^2)$	$[0,1]^3$
$F_{20}(x) = -\sum_{i=1}^{4} c_i \exp(-\sum_{j=1}^{6} a_{ij}(x_j - P_{ij})^2)$	$[0,1]^6$
$F_{21}(x) = -\sum_{i=1}^{5} [(X - a_i)(X - a_i)^T + 6c_i]^{-1}$	$[0,10]^4$
$F_{22}(x) = -\sum_{i=1}^{7} [(X - a_i)(X - a_i)^T + 6c_i]^{-1}$	$[0,10]^4$
$F_{23}(x) = -\sum_{i=1}^{10} [(X - a_i)(X - a_i)^T + 6c_i]^{-1}$	$[0,10]^4$

5.2. Algorithm for comparison

The performance of FOA has been compared with eight optimization algorithms: Genetic Algorithm (GA) [11] , Particle Swarm Optimization (PSO) [14], Gravitational Search Algorithm (GSA) [28], Teaching–Learning-Based Optimization (TLBO) [29], Grey Wolf Optimizer (GWO) [30], whale optimization algorithm (WOA) [31],Grasshopper Optimization Algorithm (GOA) [32] and Emperor Penguin Optimizer (EPO) [33].

5.3. Performance comparison

Unimodal high-dimensional functions: Average of results obtained during 20 runs of the algorithm are shown in Table 4. Functions F_1 to F_7 in Table 4 are considered as monostatic functions. These results show that FOA has a better performance in comparison to other optimization algorithm.

Multimodal high-dimensional functions: In multi-static functions like functions F_8 to F_{13}, Table 5, by increasing functions dimensions local responses are increased exponentially. Thus, finding their minimum responses is highly difficult. In these functions, arriving at the answer close to the desired response is inferred as the algorithm's efficiency in passing through local wrong responses. Results obtained by assessing functions F_8 to F_{13} after 20 times running of algorithms are shown in Table 5. Among all these algorithms FOA has a better performance. ***Multimodal low-dimensional functions:*** Functions F_{14} to F_{23}, Table 6, have the lowest number of dimensions and the lowest number of local responses. Results obtained after running algorithms for 20 times are shown in Table 6. These results well depict the FOA better performance in comparison to other optimization algorithm.

Table 4. Minimization results of benchmark functions in Table 1.

		GA	PSO	GSA	TLBO	GWO	WOA	GOA	EPO	FOA
	Average best-so-far	23.13	1.8E-03	3.75E-08	3.51E-08	2.93E-08	3.21E-08	2.54E-08	3.09E-08	**2.3E-09**
F_1	Median best-so-far	21.87	1.2E-03	1.53E-09	1.33E-09	1.03E-09	1.14E-09	9.07E-10	1.16E-09	**8.86E-12**
	Average mean fitness	23.45	5.0E-02	1.37E-05	1.29E-05	9.22E-06	1.12E-05	9.27E-06	9.75E-06	**8.1E-08**
	Average best-so-far	1.07	2.0	6.81E-05	5.77E-05	4.89E-05	5.81E-05	4.48E-05	5.31E-05	**3.94E-06**
F_2	Median best-so-far	1.13	1.9E-03	2.70E-06	2.44E-06	2.39E-06	2.40E-06	2.13E-06	2.40E-06	**2E-07**
	Average mean fitness	1.07	2.0	7.21E-04	6.41E-04	5.14E-04	5.72E-04	4.88E-04	5.65E-04	**4.6E-05**
	Average best-so-far	5.6E03	4.1E03	515.8255	474.3833	376.0085	434.5015	327.185795	428.5135	**295.381**
F_3	Median best-so-far	5.6E03	2.2E03	310.8519	273.7557	211.9048	244.2885	154.51697	222.0258	**56.93836**
	Average mean fitness	5.6E03	2.9E03	596.5089	506.003	422.5738	460.8487	413.335365	467.3401	**399.656**
	Average best-so-far	11.78	8.1	1.205542	1.135605	0.875257	1.080866	0.858736823	0.993865	**0.785953**
F_4	Median best-so-far	11.94	7.4	0.539036	0.543148	0.454636	0.529933	0.446200961	0.525235	**0.402746**
	Average mean fitness	11.78	23.6	8.759519	7.554117	5.870374	6.923197	5.571285788	6.903329	**5.210601**
	Average best-so-far	1.1E03	3.6E04	111.0475	101.8052	85.70288	99.36905	71.00497787	100.7185	**62.8803**
F_5	Median best-so-far	1.0E03	1.7E03	36.65635	33.06776	24.395652	29.83538	17.457305473	26.15271	**6.9483**
	Average mean fitness	1.1E03	3.7E04	149.2912	146.3147	136.6725	139.6439	122.2325338	140.6996	**109.1664**
	Average best-so-far	24.01	1.0E-03	1.17E-04	1.12E-05	9.18E-8	1.08E-06	8.7960E-12	9.47E-07	**0**
F_6	Median best-so-far	24.55	6.6E-03	8.17E-05	7.47E-06	5.58E-8	7.05E-07	5.6426E-13	6.20E-07	**0**
	Average mean fitness	24.52	0.02	9.39E-03	9.34E-01	7.30E-03	8.75E-02	6.6251E-5	8.19E-02	**0**
	Average best-so-far	0.06	0.04	0.036461	0.034787	0.026285	0.030744	0.025915398	0.031013	**0.0229**
F_7	Median best-so-far	0.06	0.04	0.024294	0.021833	0.0152	0.020184	0.008077661	0.017521	**0.0079**
	Average mean fitness	0.56	1.04	1.723781	1.471224	1.0152	1.250494	0.884298485	1.148805	**0.8657**

Table 5. Minimization results of benchmark functions in Table 2.

		GA	PSO	GSA	TLBO	GWO	WOA	GOA	EPO	FOA
F_8	Average best-so-far	-1.2E4	-9.8E03	-7649.61	-8983.16	-12339.5	-11189.8	-12433.3	-10023.1	**-12564.1**
	Median best-so-far	-1.2E4	-9.8E03	-7596.7	-8927.64	-12421.6	-11215.5	-12515.8	-10030.5	**-12569.5**
	Average mean fitness	-1.2E4	-9.8E03	-7625.17	-8963.13	-12387.2	-11235.2	-12483.2	-10049.4	**-12501.1**
F_9	Average best-so-far	5.90	55.1	2.00E-04	4.02E-05	1.74E-06	9.65E-06	7.21E-07	2.89E-06	**5.67E-07**
	Median best-so-far	5.71	55.6	3.80E-08	3.85E-09	6.72E-11	8.58E-10	5.74E-11	2.66E-10	**4.76E-11**
	Average mean fitness	5.92	72.8	5.897	6.5655	7.808032	8.7214	4.51464	7.16236	**2.3744**
F_{10}	Average best-so-far	2.13	9.0E-03	7.24E-04	4.64E-04	1.48E-05	1.38E-04	1.05E-05	3.90E-05	**7.64E-06**
	Median best-so-far	2.16	6.0E-03	3.91E-04	7.38E-05	1.94E-06	1.23E-05	1.23E-06	3.93E-06	**8.64E-07**
	Average mean fitness	2.15	0.02	4.53E-02	4.31E-03	1.19E-04	8.56E-04	9.37E-05	2.73E-04	**8.31E-05**
F_{11}	Average best-so-far	1.16	0.01	0.229344	0.659285	0.078421	0.271735	0.036649	0.09272	**0.0329**
	Median best-so-far	1.14	0.0081	2.44E-04	1.66E-04	3.44E-04	2.75E-04	1.41E-04	9.35E-04	**8.26E-10**
	Average mean fitness	1.16	0.055	0.6753	0.18194	0.542009	0.570567	0.95923	0.169546	**000214**
F_{12}	Average best-so-far	0.051	0.29	1.20E-10	1.12E-10	1.86E-11	1.11E-10	7.79E-12	4.26E-11	**4.59E-12**
	Median best-so-far	0.039	0.11	2.71E-11	2.83E-12	1.03E-13	6.69E-13	4.58E-14	2.46E-13	**2.35E-14**
	Average mean fitness	0.053	9.3E03	1.82E-09	2.64E-10	2.55E-11	1.12E-10	2.07E-11	4.13E-11	**1.51E-11**
F_{13}	Average best-so-far	0.081	8.291E-4	4.79E-08	1.40E-08	6.30E-10	4.62E-09	2.55E-10	1.36E-09	**1.4E-10**
	Median best-so-far	0.032	2.463E-7	9.47E-10	1.23E-10	2.71E-12	1.98E-11	1.46E-12	7.29E-12	**1.24E-12**
	Average mean fitness	0.081	6.251E-2	6.71E-05	2.38E-05	1.03E-06	4.78E-06	5.51E-07	2.42E-06	**3.26E-7**

Table 6. Minimization results of benchmark functions in Table 3.

		GA	PSO	GSA	TLBO	GWO	WOA	GOA	EPO	FOA
F_{14}	Average best-so-far	0.998	0.998	0.998	0.998	0.998	0.998	0.998	0.998	**0.998**
	Median best-so-far	0.998	0.998	0.998	0.998	0.998	0.998	0.998	0.998	**0.998**
	Average mean fitness	0.998	0.998	0.998	0.998	0.998	0.998	0.998	0.998	**0.998**
F_{15}	Average best-so-far	4.0E-03	2.8E-03	9.12E-04	9.17E-04	9.21E-04	9.31E-04	9.35E-04	9.26E-04	**9.4E-04**
	Median best-so-far	1.7E-03	7.1E-04	3.20E-04	3.22E-04	3.23E-04	3.27E-04	3.28E-05	3.25E-04	**3.3E-05**
	Average mean fitness	4.0E-03	215.6	1.52E-01	1.53E-01	1.54E-01	1.55E-01	1.56E-01	1.54E-01	**5.2-E03**
F_{16}	Average best-so-far	-1.0313	-1.03136	-1.03136	-1.03136	-1.03136	-1.03136	-1.03136	-1.03136	**-1.03163**
	Median best-so-far	-1.0313	-1.03136	-1.03136	-1.03136	-1.03136	-1.03136	-1.03136	-1.03136	**-1.03163**
	Average mean fitness	-1.0313	-1.03136	-1.03136	-1.03136	-1.03136	-1.03136	-1.03136	-1.03136	**-1.03136**
F_{17}	Average best-so-far	0.3996	0.3979	0.3979	0.3979	0.3979	0.3979	0.3979	0.3979	**0.3979**
	Median best-so-far	0.3980	0.3979	0.3979	0.3979	0.3979	0.3979	0.3979	0.3979	**0.3978**
	Average mean fitness	1.1696	2.4112	0.9526	0.8965	0.4015	05264	0.4125	0.4165	**0.3999**
F_{18}	Average best-so-far	5.70	3.0	3.0	3.0	3.0	3.0	3.0	3.0	**3.0**
	Median best-so-far	3.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0	**3.0**
	Average mean fitness	5.70	3.0	3.0	3.0	3.0	3.0	3.0	3.0	**3.0**
F_{19}	Average best-so-far	-3.8627	-3.8628	-3.8628	-3.8628	-3.8628	-3.8628	-3.8628	-3.8628	**-3.8628**
	Median best-so-far	-3.8628	-3.8628	-3.8628	-3.8628	-3.8628	-3.8628	-3.8628	-3.8628	**-3.8628**
	Average mean fitness	-3.8627	-3.8628	-3.8628	-3.8628	-3.8628	-3.8628	-3.8628	-3.8628	**-3.8628**
F_{20}	Average best-so-far	-3. 2823	-3.2369	-3.26538	-3.28169	-3.26889	-3.29813	-3.29299	-3.28005	**-3.3099**
	Median best-so-far	-3.3217	-3.2531	-3.27472	-3.29111	-3.2809	-3.3075	-3.29751	-3.28588	**-3.3320**
	Average mean fitness	-3. 2704	-3.2369	-3.25648	-3.27271	-3.26879	-3.28913	-3.27929	-3.26808	**-3.3098**
F_{21}	Average best-so-far	-5.6605	-6.6290	-5.93613	-5.90225	-6.02935	-5.9699	-6.16590	-5.99554	**-6.22713**
	Median best-so-far	-2.6824	-5.1008	-6.96062	-6.92589	-7.06551	-6.99546	-7.10096	-7.03038	**-10.1532**
	Average mean fitness	-5.6605	-5.7496	-5.63615	-5.60805	-5.72103	-5.66421	-5.74972	-5.69266	**-6.11845**
F_{22}	Average best-so-far	-7.3421	-8.1118	-7.95206	-7.91237	-8.07163	-7.99172	-8.1121	-8.03153	**-9.03047**
	Median best-so-far	-10.3932	-10.402	-10.1963	-10.1456	-10.3502	-10.2474	-10.4022	-10.2989	**-10.4029**
	Average mean fitness	-7.3421	-7.9305	-7.77376	-7.73514	-7.89114	-7.81259	-7.93066	-7.85179	**-9.83367**
F_{23}	Average best-so-far	-6.2541	-6.7634	-6.62987	-6.59689	-6.72983	-6.66315	-6.76354	-6.6965	**-9.71723**
	Median best-so-far	-4.5054	-10.536	-10.3277	-10.2762	-10.4838	-10.3793	-10.5361	-10.4314	**-10.5364**
	Average mean fitness	-6.2541	-6.7626	-6.62959	-6.59674	-6.72915	-6.66262	-6.76269	-6.6958	**-8.59024**

5.4. FOA in power system application

Optimization algorithms such as FOA have many applications in various sciences, especially power engineering. One can mention the following: Placement of Distributed Generation [34], Placement of battery energy storage [35], FACTS devices [36], Microgrid [37], Protection [38],Energy carrier [39] and operation of power plants [40] .In the next section, the placement of distributed generation is studied using FOA.

5.5. Placement of Distributed Generation (DGs)

The distribution system planning requires DGs to be placed properly within the distribution system. In definition, DG known as a small generator is responsible of generating Stand Alone and On Grid electricity.[41]. In placing DGs, some methods can be used. [42, 43] refers to the sizing and analytical method by which DGs can be placed and sized properly. The objective function of the mentioned reference is minimizing the loss.

Minimizing the overall system loss and improving the system voltage profile are indeed two principal objectives of DGs optimal sizing and placement. The objective function is shown in equation (5).

$$objective\ function = K_1 Loss + K_2 \sum_{i=1}^{N_B} |V_i - 1| \quad (5)$$

Where K_1 and K_2 is weight coefficients, $Loss$ is overall system loss, N_B is the number of buses and V_i is the voltage of buses.

In order to simulate the proposed problem, the IEEE 33-bus radial network is used. The networks data, including the resistance and reactance of the lines and the loads connected to nodes, were presented in [44].

placement and sizing of DG units regarding the minimum value of the problem objective function are defined. The results for this case are shown in Table 7. It shows the effectiveness of FOA for finding the optimal placement and sizing of DGs. The obtained results are assessed comparing with other algorithm. FOA gives a significant reduction in total active power loss to be 71.052 kW with reduction of 64.9% referred to initial case. Three DGs are installed at buses 14, 24, and 29 with penetration 0.8547, 1.1018, and 1.1812 MW, respectively. The minimum voltage level (0.974) is obtained at bus 33.

Table 7. placement and sizing of DGs.

Method	Power loss (kW)	DG size (MW) and location	Min. voltage (p.u.)
GA	108.41	1.2461 (11), 0.2156 (18), 1.2145 (33)	0.981(29)
PSO	103.22	0.9250 (11), 0.8630 (16), 1.2 (32)	0.980 (25)
GSA	106.28	1.50 (11), 0.4228 (29), 1.0714 (30)	0.982(25)
TLBO	105.34	1.1768 (8), 0.9816 (13), 0.8297 (32)	0.980(30)
GWO	91.25	0.5876 (15), 0.1959 (25), 0.783 (33)	0.956(33)
WOA	98.3	0.633 (17), 0.09 (18), 0.9470 (33)	0.963 (25)
GOA	96.75	0.5724 (17), 0.107 (18), 1.0462 (33)	0.965 (29)
EPO	88.65	0.5897 (14), 0.189 (18), 1.0146 (32)	0.965 (32)
FOA	**71.05**	0.8547 (14), 1.1018 (24), 1.1812 (29)	0.974 (33)

6. CONCLUSIONS

A new optimization algorithm, called the Following Optimization Algorithm (FOA) is introduced in this paper. FOA is characterized based on the community members' social standing. In a given community, people work to improve their community and make it ideal by following each other. FOA is a simple algorithm in which people in a society follow, consistent with the following co-efficient *f* in each iteration of the algorithm, a person who has the highest social status in the society. This parameter is set to have a better exploration and exploitation efficiency. To assess its performance, the proposed algorithm is tested for a set of standard benchmark test functions. In addition, the application of FOA was investigated on the optimal placement and sizing of DGs in power engineering. Results show that FOA has a better performance than the GA, PSO, GSA, TLBO, GWO, WOA, GOA and EPO algorithms.

REFERENCES

[1] M. Dehghani, Z. Montazeri, A. Dehghani, N. Nouri, and A. Seifi, "BSSA: Binary spring search algorithm," *Proce. IEEE 4th Int. Conf. Knowl. Base. Eng. Innovation (KBEI).*, 2017, pp. 0220-0224.

[2] M. Dehghani, Z. Montazeri, A. Dehghani, and A. Seifi, "Spring search algorithm: A new meta-heuristic optimization algorithm inspired by Hooke's law," *Proce. IEEE 4th Int. Conf. Knowl. Base. Eng. Innovation (KBEI).*, 2017, pp. 0210-0214.

[3] M. Dehghani, Z. Montazeri, O. P. Malik, A. Ehsanifar, and A. Dehghani, "OSA: Orientation Search Algorithm," *Int. J. Ind. Electron., Control. Optim.*, vol. 2, pp. 99-112, 2019.

[4] M. Dehghani, M. Mardaneh, O. P. Malik, and S. M. NouraeiPour, " DTO: Donkey Theorem Optimization," *Proce. 2019 Iran. Conf. Electr. Eng. (ICEE).*, Yazd, Iran, 2019.

[5] M. Bielli and P. Carotenuto, "Genetic algorithms and transportation analysis: review and perspectives for bus network optimization," *N. Anal. Adv. Transp. Spatial Dyn.*, 2018, pp. 35-48.

[6] I. V. Antonov, E. Mazurov, M. Borodovsky, and Y. A. Medvedeva, "Prediction of lncRNAs and their interactions with nucleic acids: benchmarking bioinformatics tools," *Briefings Bioinf.*, vol. 20, pp. 551-564, 2018.

[7] Z. Montazeri and T. Niknam, "Energy carriers management based on energy consumption," *Proce. IEEE 4th Int. Conf. Knowl. Base. Eng. Innovation (KBEI).*, 2017, pp. 0539-0543.

[8] S. Darby, T. V. Mortimer-Jones, R. L. Johnston, and C. Roberts, "Theoretical study of Cu–Au nanoalloy clusters using a genetic algorithm," *J. Chem. Phys.*, vol. 116, pp. 1536-1550, 2002.

[9] S. Dehbozorgi, A. Ehsanifar, Z. Montazeri, M. Dehghani, and A. Seifi, "Line loss reduction and voltage profile improvement in radial distribution networks using battery energy storage system," *Proce. IEEE 4th Int. Conf. Knowl. Base. Eng. Innovation (KBEI).*, 2017, pp. 0215-0219.

[10] C. A. C. Coello, E. H. Luna, and A. H. Aguirre, "Use of particle swarm optimization to design combinational logic circuits," *Proce. Int. Conf. Evolvable Syst.*, 2003, pp. 398-409.

[11] K.-S. Tang, K.-F. Man, S. Kwong, and Q. He, "Genetic algorithms and their applications," *IEEE Signal Process. Mag.*, vol. 13, pp. 22-37, 1996.

[12] J. D. Farmer, N. H. Packard, and A. S. Perelson, "The immune system, adaptation, and machine learning," *Phys. Nonlinear Phenom.*, vol. 22, pp. 187-204, 1986.

[13] M. Dorigo and L. M. Gambardella, "Ant colony system: a cooperative learning approach to the traveling salesman problem," *IEEE Trans. Evol. Comput.*, vol. 1, pp. 53-66, 1997.

[14] J. Kenny, "Particle swarm optimization," *Proc. 1995 IEEE Int. Conf. Neural Networks*, 1995, pp. 1942-1948.

[15] K. E. Voges, N. Pope, and M. R. Brown, "Cluster analysis of marketing data examining on-line shopping orientation: A comparison of k-means and rough clustering approaches," *Heuristics Optim. Knowl. Discovery*, pp. 207-224, 2002.

[16] A. Lazar, "Heuristic knowledge discovery for archaeological data using genetic algorithms and rough sets," *Heuristics Optim. Knowl. Discovery*, pp. 263-278, 2002.

[17] G. Gigerenzer and W. Gaissmaier, "Heuristic decision making," *Annu. Rev. Psychol.*, vol. 62, pp. 451-482, 2011.

[18] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Sci.*, vol. 220, pp. 671-680, 1983.

[19] Z. W. Geem, J. H. Kim, and G. Loganathan, "A new heuristic optimization algorithm: harmony search," *Simul.*, vol. 76, pp. 60-68, 2001.

[20] S. A. Hofmeyr and S. Forrest, "Architecture for an artificial immune system," *Evol. Comput.*, vol. 8, pp. 443-473, 2000.

[21] M. Dorigo and T. Stützle, "Ant colony optimization: overview and recent advances," *Handb. metaheuristics*, ed: Springer, 2019, pp. 311-351.

[22] K. M. Passino, "Biomimicry of bacterial foraging for distributed optimization and control," *IEEE Control Syst. Mag.*, vol. 22, pp. 52-67, 2002.

[23] A. Silva, A. Neves, and E. Costa, "An empirical comparison of particle swarm and predator prey optimisation," *Proce. Ir. Conf. Artif. Intell. Cognit. Sci.*, 2002, pp. 103-110.

[24] R. A. Formato, "Central force optimization: a new metaheuristic with applications in applied electromagnetics," *Prog. Electromagnet. Res.*, vol. 77, pp. 425-491, 2007.

[25] P. Tarasewich and P. R. McMullen, "Swarm intelligence: power in numbers," *Commun. ACM.*, vol. 45, pp. 62-67, 2002.

[26] A. E. Eiben and C. A. Schippers, "On evolutionary exploration and exploitation," *Fundam. Inf.*, vol. 35, pp. 35-50, 1998.

[27] X. Yao, Y. Liu, and G. Lin, "Evolutionary programming made faster," *IEEE Trans. Evol. Comput.*, vol. 3, pp. 82-102, 1999.

[28] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, "GSA: a gravitational search algorithm," *Inf. Sci.*, vol. 179, pp. 2232-2248, 2009.

[29] P. Sarzaeim, O. Bozorg-Haddad, and X. Chu, "Teaching-Learning-Based Optimization (TLBO) Algorithm," *Adv. Optim.* Nat. *Inspired Algorithms*, ed: Springer, 2018, pp. 51-58.

[30] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Software*, vol. 69, pp. 46-61, 2014.

[31] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Adv. Eng. Software*, vol. 95, pp. 51-67, 2016.

[32] S. Saremi, S. Mirjalili, and A. Lewis, "Grasshopper optimisation algorithm: theory and application," *Adv. Eng. Software*, vol. 105, pp. 30-47, 2017.

[33] G. Dhiman and V. Kumar, "Emperor penguin optimizer: A bio-inspired algorithm for engineering problems," *Knowl. Based Syst.*, vol. 159, pp. 20-50, 2018.

[34] M. Dehghani, M. Mardaneh, Z. Montazeri, A. Ehsanifar, M. J. Ebadi, and O. M. Grechko, "Spring search algorithm for simultaneous placement of distributed generation and capacitors," *Electr. Eng. Electrochem.*, vol. 6, pp. 68-73, 2018.

[35] R. Afshan and J. Salehi, "Optimal scheduling of battery energy storage system in distribution network considering uncertainties using hybrid Monte Carlo-Genetic approach," *J. Oper. Autom. Power Eng.*, vol. 6, pp. 1-12, 2018.

[36] Z. Luburić and H. Pandžić, "FACTS devices and energy storage in unit commitment," *Int. J. Electr. Power Energy Syst.*, vol. 104, pp. 311-325, 2019.

[37] K. Mazlumi and A. Shabani, "DC microgrid protection in the presence of the photovoltaic and energy storage systems," *J. Oper. Autom. Power Eng.*, vol. 6, pp. 243-254, 2018.

[38] A. Ehsanifar, M. Dehghani, and M. Allahbakhshi, "Calculating the leakage inductance for transformer inter-turn fault detection using finite element method," *Proce. 2017 Iran. Conf. Electr. Eng. (ICEE)*, 2017, pp. 1372-1377.

[39] M. Dehghani, Z. Montazeri, A. Ehsanifar, A. R. Seifi, M. J. Ebadi, and O. M. Grechko, "Planning of energy carriers based on final energy consumption using dynamic programming and particle swarm optimization," *Electr. Eng. Electrochem.*, vol. 5, pp. 62-71, 2018.

[40] Z. Montazeri and T. Niknam, "Optimal utilization of electrical energy from power plants based on final energy consumption using gravitational search algorithm," *Electr. Eng. Electrochem.*, vol. 4, pp. 70-73, 2018.

[41] P. A. Daly and J. Morrison, "Understanding the potential benefits of distributed generation on power delivery systems," *Proce. Rural Electr. Power Conf., 2001*, 2001, pp. A2/1-A213.

[42] T. Gözel and M. H. Hocaoglu, "An analytical method for the sizing and siting of distributed generators in radial systems," *Electr. Power Syst. Res.*, vol. 79, pp. 912-918, 2009.

[43] C. Wang and M. H. Nehrir, "Analytical approaches for optimal placement of distributed generation sources in power systems," *IEEE Trans. Power Syst.*, vol. 19, pp. 2068-2076, 2004.

[44] M. Ettehadi, H. Ghasemi, and S. Vaez-Zadeh, "Voltage stability-based DG placement in distribution networks," *IEEE Trans. Power Delivery*, vol. 28, pp. 171-178, 2012.