Education System Search: A New Population-based Metaheuristic Optimization Algorithm

Hossein Moradi[1*], Hossein Ebrahimpour-Komleh[2]
1- Department of Computer Engineering, Faculty of Computer and Electrical Engineering, University of Kashan, Kashan, Iran.
E-mail: moradyhsnm@yahoo.com (Corresponding author)
2- Department of Computer Engineering, Faculty of Computer and Electrical Engineering, University of Kashan, Kashan, Iran.
E-mail: ebrahimpour@kashanu.ac.ir

Received: January 2019 Revised: April 2019 Accepted: June 2019

ABSTRACT:
Optimization algorithms inspired by nature as intelligent optimization methods with classical methods have demonstrated significant success. Some of these techniques are genetic algorithms, inspired by biological evolution of humans and other creatures) ant colony optimization and simulated annealing method (inspired by the refrigeration process metals). The methods for solving optimization problems in many different areas such as determining the optimal course of their work, designing optimal control for industrial processes, solving industrial engineering major issues such as the optimal layout design for industrial units, problem solving, and queuing in the design of intelligent agents have been used. This paper introduces a new algorithm for optimization, which is not a natural phenomenon, but a phenomenon inspired teaching-human. It is entitled Education System algorithm (ESA). Results demonstrate this method is better than other method in this area.

KEYWORDS: Education System Algorithm, Metaheuristic, Optimization, Evolutionary Algorithm.

1. INTRODUCTION

Metaheuristic algorithms are precise algorithms employed to find the optimal solution. Methods and optimization algorithms are classified into two categories: exact algorithms and approximation algorithms. Approximation algorithms are able to find good solutions (near optimal) within a short time for hard optimization problems. Approximation algorithms are classified into three categories: heuristic, metaheuristic and hyper-heuristic algorithms, which are parts of a package. The main problem with heuristic algorithms is placing them into local optimization, and their inability to use the various issues. Metaheuristic algorithms solve the problems represented as heuristic algorithms. The meta-heuristic algorithms and optimization algorithms are approximate one of the strategies that have been out of local optimum and applicable for a wide range of related issues. Various classes of these types of algorithms have been developed in the recent decades [4].

There are different criteria for categorizing metaheuristic algorithms:

1. single-solution-based and population-based: algorithms based on a solution in the investigation of a solution during the process of change, while in the population during search-based algorithms, some solutions are considered.

2. Inspired by nature and inspired without nature: a large number of metaheuristic algorithms are inspired by nature, but some metaheuristic algorithms may not have been inspired by nature.

3. Memory based and without memory based: Some metaheuristic algorithms suffer lack of memory, in the sense that, these algorithms do not use the information obtained during the search (for example, simulated annealing). However, in some metaheuristic search, algorithms such as prohibited algorithms are used based on memory. The memory stores the information obtained during the research.

4. Deterministic and probable: some metaheuristic algorithms use deterministic decisions for solving the problems [5]. However, metaheuristic algorithms may simulate annealing where a set of rules might be used during this research [6].

Almost all metaheuristics share the following characteristics: they are nature-inspired (based on some principles from physics, biology, and ethology); they make use of stochastic components (involving random variables); they do not use the gradient or Hessian matrix of the objective function; they have several parameters which should fit the presented problem [7, 8].

One of the interesting branches of the population-based meta-heuristics is Swarm Intelligence (SI). SI is the emergent collective intelligence of groups of simple agents. The inspirations of SI techniques originate mostly from natural colonies, flock, herds, and schools. Some of the most popular SI techniques include ACO, PSO, and Artificial Bee Colony (ABC)[8].

Regardless of the differences between the meta-heuristics, a common feature is the division of the search process into two phases: exploration and exploitation. The exploration phase refers to the process of investigating the promising area(s) of the search space as broadly as possible. An algorithm needs to have stochastic operators to randomly and globally search the search space in order to support this phase. On the other hand, exploitation refers to the local search capability around the promising regions obtained in the exploration phase. Finding a proper balance between these two phases is considered a challenging task due to the stochastic nature of meta-heuristics. This paper proposes a new SI technique inspired by educational systems.

2. PROPOSED ALGORITHM

This method is based on the educational system and it is designed for universities or the private sectors. At first, some students study in this system, classified into several classes. Professors and assistant professors are training individuals. Professors and assistant professors are the best choice. During the test, after training the instructor and assistant professor, the best choices of each class are selected. In the next step, leaders of academics go to the next step. In this stage the, best teacher of all teachers will be selected to instruct other teachers. Again, during the exam for answering the first stage, the best teacher will be introduced. The process to achieve the best results as a final condition is metaheuristic algorithms. The flowchart of this algorithm is presented in Fig. 1 in which the Supplementary Description can be observed.

Algorithm:

1. Select students, professors, assistant professors, and allocate students to professors and assistant professors.

Some random points in the research space as well as the primary classes should be selected. Initially, specify a cost function and calculate the cost of selected points. Costs of points (students) are arranged in an ascending order. If we want to build n class, initially we should select the best n of points as n professors of class. It is followed by the best n so the second points serve as n assistant professors. The rest of students claim approximately equal size in each class.

1. Educate students by professor and assistant professor

In this stage, the professor and assistant professor starts teaching their students. This means that these students will be allocated to the professors and their assistants.

$$x_{new} = x_{old} + \alpha . r_1 (t_1 - x_{old}) + \beta . r_2 (t_2 - x_{old}) \quad \alpha > \beta \qquad (1)$$

The variable of t_1 is professor, t_2 is assistance professors, x_{old} represents the current position of student, and x_{new} denotes the new position of student. α and β are two positive parameters referring to the rate of teaching professor and assistant professor respectively. It is clear that the rate of professor teaching is more than that of the assistant professor. The value of these two parameters lies between one and two. In this stage, some lazy students do not move toward their professors and assistant professors. Therefore, a random value is assigned to these students. It is same as motivation in genetic algorithm.

3. REPLACEMENT

In this stage, all students, their professors and assistant professors are evaluated by the cost function. Then, if there are more best students than their professors and assistant professors, they should be replaced.

4. Find the best class

In this stage, the best class and consequently the best professors are found. Calculation of the value of the objective function for a class is as follows.

$$F_{Class} = F_{professor} + £1 F_{assistanceprofessor} + £2 mean(F_{students}) \qquad (2)$$

The experiment reveals that the best value for £1 and £2 lies between 0 and 0.2. Also the value of £1should be larger than £2.

5. Teaching the best professor to other professors

The best professor presents a seminar to the other professors. This means that other professors should move to the best professor. However, since professors hardly accept the problem, they move to the place of the best professors.

$$y_{new} = y_{old} + r.(t_3 - y_{old}) * \cos(\theta) \qquad (3)$$

The variable of t_3 is the position of the best professor and y represents the current position of every professors. Also, y_{new} is the new position of professors and θ is the place of each professor in which they are the best professor calculated randomly.

6. **Select best candidate**

In this stage the best candidate will be reported. If conditions be satisfied, the algorithm can be stop and go forward to step 7, otherwise it go to the second step and the next iteration begins.

The flowchart of this algorithm is shown in Fig. 1.

7. **End**

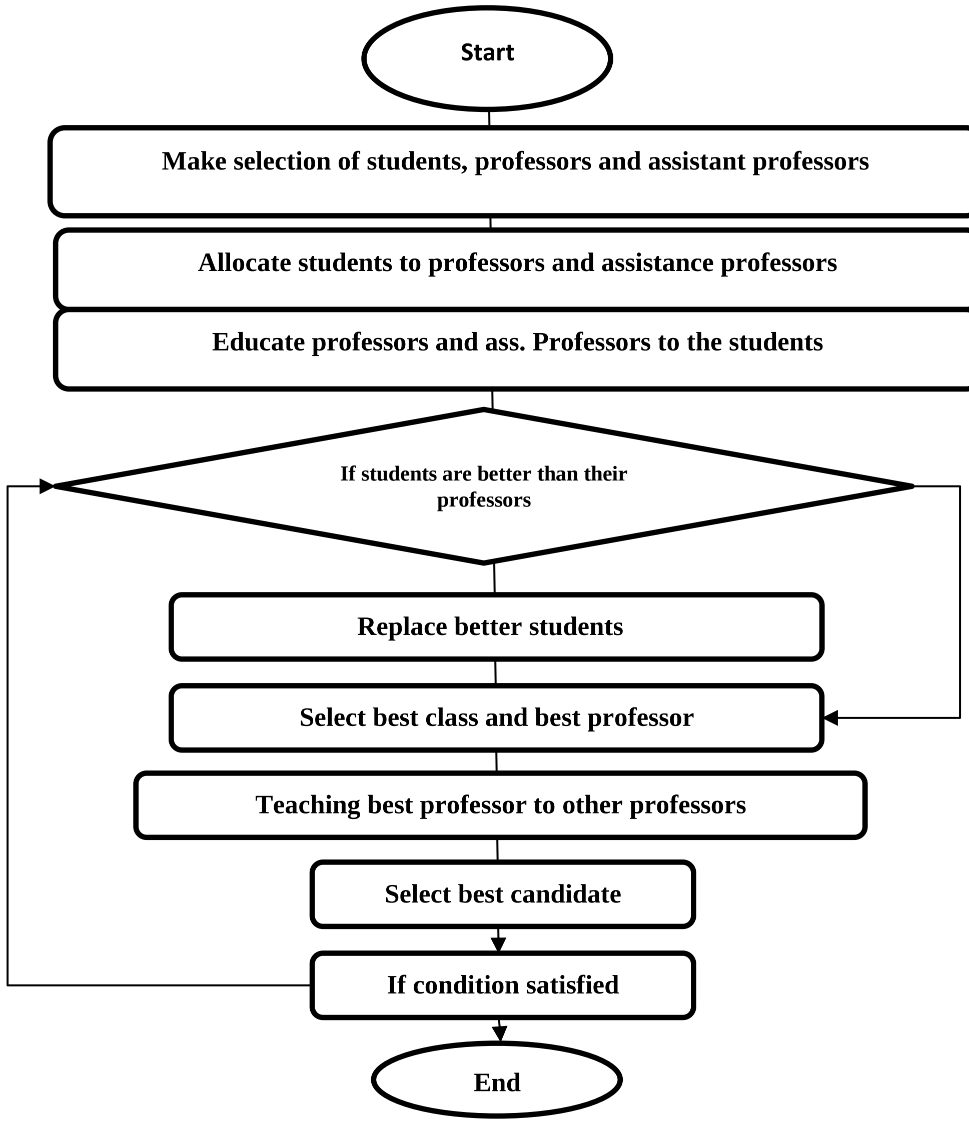
Start
Make selection of students, professors and assistant professors
Allocate students to professors and assistance professors
Educate professors and ass. Professors to the students
If students are better than their professors
Replace better students
Select best class and best professor
Teaching best professor to other professors
Select best candidate
If condition satisfied
End

Fig. 1. Flowchart of education system algorithm.

Parameters

The parameter α specifies the impact factor of professor and β parameter determines the impact factor of assistant professor. It is clear that the impact factor of professor should be larger than impact factor of assistant professor. These two parameters should be between 0 and 2. These should be multiplied by a number between 0 and 1 based on Gaussian distribution. Experiments reveal that 0.5 is suitable for β, and 2 is suitable for α. The £1 and £2 were used for calculating the best class; £1 reveals the role of assistance professor and £2 refers to the role of students. The value of £1 should be lower than £2. Experiments reveal that 0.1 is suitable for £1 and 0.05 is suitable for £2. Also, in this algorithm, there are some students and professors who do not obey their teachers and do not move toward to their teachers. The percentage of these students and professors is approximately five. This process is the same as motivation in genetic algorithm.

3. EXPERIMENTS

In this section, the ESA algorithm is benchmarked on 23 bench-mark functions. These 23 benchmark functions are the classical functions utilized by many researchers [9-14]. Despite the simplicity, we have chosen these test functions to be able to compare our results to those of the current meta-heuristics. These benchmark functions are listed in Tables 1–3 where Dim represents the dimension of the function, Range is the boundary of the function's search space, and f_{min} is the optimum. The shapes of these test functions are illustrated in Figures 2-4. Generally speaking, the benchmark functions used are minimization functions and can be divided into three groups: unimodal, multimodal, and fixed-dimension multimodal functions. Note that a detailed description of the composite benchmark functions is available in the CEC 2005 technical report [17].

The ESA algorithm was run 30 times on each benchmark function. The statistical results (average and standard deviation) are reported in Tables 4–6. For verifying the results, the ESA algorithm is compared to PSO [18] as an SI-based technique and GSA [19] as a physics-based algorithm. In addition, the ESA algorithm is compared with three EAs: DE [20], fast evolutionary programing (FEP)[21], and gray wolf optimization (GWO) [22]. And, three TLBO, ES, ICA algorithms have been tested[23,24,25].

According to the results of Table 4, ESA is able to provide very competitive results. This algorithm outperforms all others in F1, F2, and F7. Note that the unimodal functions are suitable for benchmarking operation. Therefore, these results indicate the superior performance of ESA in terms of operating the optimum. This is due to the proposed exploitation operators previously discussed.

In contrast to the unimodal functions, multimodal functions have many local optima with the number increasing exponentially with dimension. This makes them suitable for benchmarking the operability of an algorithm. According to the results of Tables 6 and 7, ESA is also able to provide very competitive results on the multimodal benchmark functions. This algorithm outperforms GWO, PSO, and GSA on the majority of the multimodal functions. Further, ESA shows very competitive results compared to DE and FEP and outperforms them occasionally. These results suggest that the ESA algorithm is advantageous in terms of operation.

Table 1. Unimodal benchmark function.

Function	Dim	Rang	f_{min}
$f_1(x)=\sum_{i=1}^{n} x_i^2$	30	[-100, 100]	0
$f_2(x)=\sum_{i=1}^{n}\lvert x_i\rvert+\prod_{i=1}^{n}\lvert x_i\rvert$	30	[-10, 10]	0
$f_3(x)=\sum_{i=1}^{n}\left(\sum_{j-1}^{i} x_j\right)^2$	30	[-100, 100]	0
$f_4(x)=\max_i\{\lvert x_i\rvert, 1\le i\le n\}$	30	[-100, 100]	0
$f_5(x)=\sum_{i=1}^{n-1}[100(x_{i+1}-x_i^2)^2+(x_i-1)^2]$	30	[-30, 30]	0
$f_6(x)=\sum_{i=1}^{n}([x_i+0.5])^2$	30	[-100, 100]	0
$f_7(x)=\sum_{i=1}^{n} ix_i^4+random[0,1]$	30	[-1.28, 1.28]	0

Table 2. Multimodel benchmark function.

Function	Dim	Rang	f_{min}
$f_8(x)=\sum_{i=1}^{n} x_i - x_i sin\left(\sqrt{\lvert x_i\rvert}\right)$	30	[-500, 500]	-5*418.9829
$f_9(x)=\sum_{i=1}^{n}[x_i^2-10cos(2\pi x_i)+10]$	30	[-5.12, 5.12]	0
$f_{10}(x)=-20exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right)-\exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right)+20+e$	30	[-32, 32]	0
$f_{11}(x)=\frac{1}{4000}\sum_{i=1}^{n} x_i^2-\prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right)+1$	30	[-600, 600]	0

$f_{12}(x) = \frac{\pi}{n}\{10\sin(\pi y_1) + \sum_{i=1}^{n-1}(y_i-1)^2[1+10sin^2(\pi y_{i+1}) + (y_n-1)^2]\} + \sum_{i=1}^{n} u(x_i, 10,100,4),\ y_i = 1+\frac{x_i+1}{4}, u(x_i,a,k,m) = \begin{cases} k(x_i-a)^m, & x_i > a \\ 0, & -a < x_i < a \\ k(-x_i-a)^m, & x_{i<-a} \end{cases}$	30	[-50, 50]	0
$f_{13}(x) = 0.1\{sin^2(3\pi x_i) + \sum_{i=1}^{n}(x_i-1)^2[1+sin^2(3\pi x_i+1)] + (x_n-1)^2\,[1+sin^2(2\pi x_n)]\} + \sum_{i=1}^{n} u(x_i, 5,100,4)$	30	[-50, -50]	0

Table 3. Fixed-dimension multimodal benchmark function.

Function	Dim	Rang	$\boldsymbol{f_{min}}$
$f_{14}(x) = \left(\frac{1}{500} + \sum_{j=1}^{25}\frac{1}{j+\sum_{i=1}^{2}\left(x_{i-a_{ij}}\right)^6}\right)^{-1}$	2	[-65, 65]	1
$f_{15}(x) = \sum_{i=1}^{11}\left[a_i - \frac{x1(b_i^2+b_ix_2)}{b_i^2+b_ix_3+x_4}\right]^2$	4	[-5, 5]	0.00030
$f_{16}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	2	[-5, 5]	-1.0316
$f_{17}(x) = (x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6)^2 + 10\left(1-\frac{1}{8\pi}\right)cosx_1 + 10$	2	[-5, 5]	0.398
$f_{18}(x) = [1+(x_1+x_2+1)^2(19-14x_1+3x_1^2-14x_2+6x_1x_2+3x_2^2][30+(2x_1-3x_2)^2(18-32x_1+12x_1^2+48x_2-36x_1x_2+27x_2^2)]$	2	[-2, 2]	3
$f_{19}(x) = -\sum_{i=1}^{4} c_i\exp(-\sum_{j=1}^{3} a_{ij}(x_j-p_{ij})^2)$	3	[1, 3]	-3.86
$f_{20}(x) = -\sum_{i=1}^{4} c_i\exp(-\sum_{j=1}^{6} a_{ij}(x_j-p_{ij})^2)$	6	[0, 1]	-3.32
$f_{21}(x) = -\sum_{i=1}^{5}[(X-a_i)(X-a_i)^T + c_i]^{-1}$	4	[0, 10]	-10.1532
$f_{22}(x) = -\sum_{i=1}^{7}[(X-a_i)(X-a_i)^T + c_i]^{-1}$	4	[0, 10]	-10.4028
$f_{23}(x) = -\sum_{i=1}^{10}[(X-a_i)(X-a_i)^T + c_i]^{-1}$	4	[0,10]	-10.5363

Table 4. Results of unimodal benchmark functions.

Fun.		ESA	PSO	GWO	GSA	DE	FEP	TlBO	ES	ICA
f₁	Mean	**1.4413e-31**	0.000136	6.59e-28	2.53e-16	8.2e-14	0.0057	2.91e-17	6.70e-19	2.74e-14
	Variance	**5.78e-5**	0.000202	6.34e-5	9.67e-11	5.9e-14	0.00013	2.76e-16	1.9e-18	7.39e-14
f₂	Mean	**2.3628e-18**	0.042144	7.18e-17	0.055655	1.5e-09	0.0081	8.39e-9	2.45e-12	8.42e-5
	Variance	**0.040345**	0.045421	0.029014	0.194074	9.9e-10	0.00077	2.48e-5	4.13e-7	3.24e-2
f₃	Mean	**5.2318e-8**	70.12562	3.29e-06	896.5347	6.8e-11	0.016	6.71e-4	3.65e-5	7.16e-3
	Variance	**0.00019**	22.11924	79.14958	318.9559	7.4e-11	0.014	5.17e-2	0.00012	0.00032
f₄	Mean	**5.6631e-10**	1.086481	5.61e-07	7.35487	0	0.3	0.00081	7.91e-7	0.00387
	Variance	**0.0021656**	0.317039	1.315088	1.741452	0	0.5	0.000325	2.45e-6	0.00457
f₅	Mean	**0**	96.71832	26.81258	67.54309	0	5.06	1.094564	5.12e-10	0.07244
	Variance	**0**	60.11559	69.90499	62.22534	0	5.87	0.035233	2.37e-8	0.00644

f_6	Mean	**3.0815e-33**	0.000102	0.816579	2.5e-16	0	0	9.45e-8	7.23e-15	4.25e-13
	Variance	**2.04e-8**	8.28e-05	0.000126	1.74e-16	0	0	2.67e-5	4.19e-12	5.65e-10
f_7	Mean	**0.000620**	0.122854	0.002213	0.089441	0.00463	0.1415	0.67564	0.00953	0.457677
	Variance	**0.013455**	0.044957	0.100286	0.04339	0.0012	0.3522	0.07359	0.04566	0.067423

Table 5. Results of multimodal benchmark functions.

Fun.		ESA	PSO	GWO	GSA	DE	FEP	TlBO	ES	ICA
f_8	Mean	**-3951.7941**	-4841.29	-6123.1	-2821.07	-11080	-12554.5	-5653.76	-4056.43	-7896.98
	Variance	**-2934.7621**	1152.814	-4087.44	493.0375	574.7	52.6	-3276.37	-1243.46	-2235.16
f_9	Mean	**1.1369e-13**	46.70423	0.310521	25.96841	69.2	0.046	6.94e-03	4.67e-10	0.01294
	Variance	**2.1234e-13**	11.62938	47.35612	7.470068	38.8	0.012	0.00829	2.49e-06	0.00284
f_{10}	Mean	**2.931e-14**	0.276015	1.06e-13	0.062087	9.7e-08	0.018	0.00385	53e-12	0.00432
	Variance	**0.005476**	0.50901	0.077835	0.23628	4.2e-08	0.0021	0.00034	25e-09	0.00294
f_{11}	Mean	**0.0022751**	0.009215	0.004485	27.70154	0	0.016	0.27459	0.05922	0.50382
	Variance	**0.0048224**	0.007724	0.006659	5.040343	0	0.022	0.00434	0.00354	0.02843
f_{12}	Mean	**0.0002065**	0.006917	0.053438	1.799617	7.9e-15	9.2e-6	0.03485	0.06392	0.39288
	Variance	**0.0003456**	0.026301	0.020734	0.95114	8e-15	3.6e-6	0.04222	0.00283	0.01943
f_{13}	Mean	**3.0694e-27**	0.006675	0.654464	8.899084	5.1e-14	0.00016	53e-07	84e-19	23e-05
	Variance	**2.509e-27**	0.008907	0.004474	7.126241	4.8e-14	0.000073	84e-05	42e-14	74e-03

Table 6. Results of fixed-dimension multimodal benchmark functions

Fun.		ESA	PSO	GWO	GSA	DE	FEP	TlBO	ES	ICA
f_{14}	Mean	**2.674601**	3.627168	4.042493	5.859838	0.998004	1.22	4.743503	3.862627	2.903265
	Variance	**2.672122**	2.560828	4.252799	3.831299	3.3e-16	0.56	3.196064	2.537048	1.885959
f_{15}	Mean	**0.000396**	0.000577	0.000337	0.003673	4.5e-14	0.0005	0.002125	0.001719	0.000932
	Variance	**0.000782**	0.000222	0.000625	0.001647	0.00033	0.00032	0.000935	0.000781	0.000441
f_{16}	Mean	**-1.03163**	-1.03163	-1.03163	-1.03163	-1.03163	-1.03	-1.03163	-1.03122	-1.03095

	Variance	**-1.03163**	6.25e-16	-1.03163	4.88e-16	3.1e-13	4.9e-7	5.57e-13	1.23e-05	2.04E-07
f_{17}	Mean	**0.39789**	0.397887	0.397889	0.397887	0.397887	0.398	0.397887	0.397915	0.397934
	Variance	**0.39788**	0	0.397887	0	9.9e-9	1.5e-7	0	3.75E-08	6.25E-08
f_{18}	Mean	**3**	3	3.000028	3	3	3.02	3	3.005	3.008333
	Variance	**0**	1.33e-15	3	4.17e-15	2e-15	0.11	2.75E-15	0.0275	0.045833
f_{19}	Mean	**-3.8628**	-3.86278	-3.86263	-3.86278	N/A	-3.86	-3.86278	-3.86209	-3.86162
	Variance	**2.8e-16**	2.58e-15	-3.86278	2.29e-15	N/A	0.000014	2.44e-14	3.5e-06	5.83E-06
f_{20}	Mean	**-3.2031**	3.26634	-3.28654	-3.31778	N/A	-3.27	-0.02572	-0.83679	-0.28015
	Variance	**0.003459**	0.060516	-3.25056	0.023081	N/A	0.059	0.041799	0.046099	0.055205
f_{21}	Mean	**-9.40322**	-6.8651	-10.1514	-5.95512	-10.1532	-5.52	-6.41011	-6.18758	-6.19089
	Variance	**2.03844**	3.019644	-9.14015	3.737079	0.0000025	1.59	3.378362	2.931271	2.513638
f_{22}	Mean	**-10.4029**	-8.45653	-10.4015	-9.68447	-10.4029	-5.53	-9.0705	-8.18538	-7.39064
	Variance	**0**	3.087094	-8.58441	2.014088	3.9E-7	2.12	2.550591	2.442943	2.550012
f_{23}	Mean	**-10.5364**	-9.95291	-10.5343	-10.5364	-10.5364	-6.57	-10.2447	-9.32599	-8.6163
	Variance	**0**	1.782786	-8.55899	2.6e-15	1.9E-07	3.14	0.891393	1.453545	2.125444

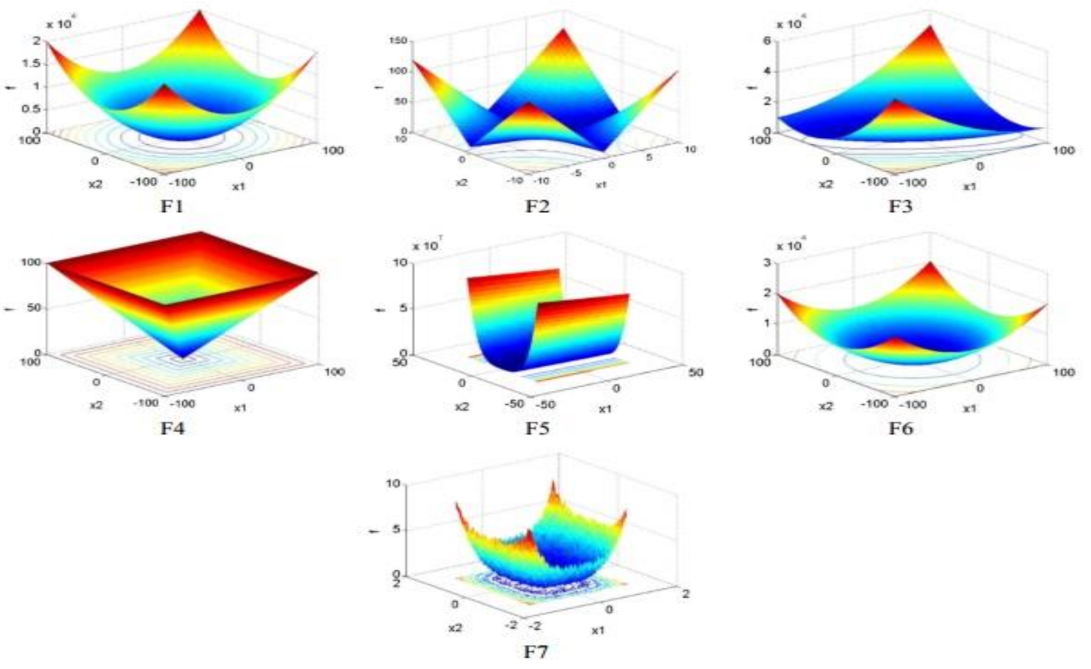

Fig. 2. Search space of unimodal test functions.

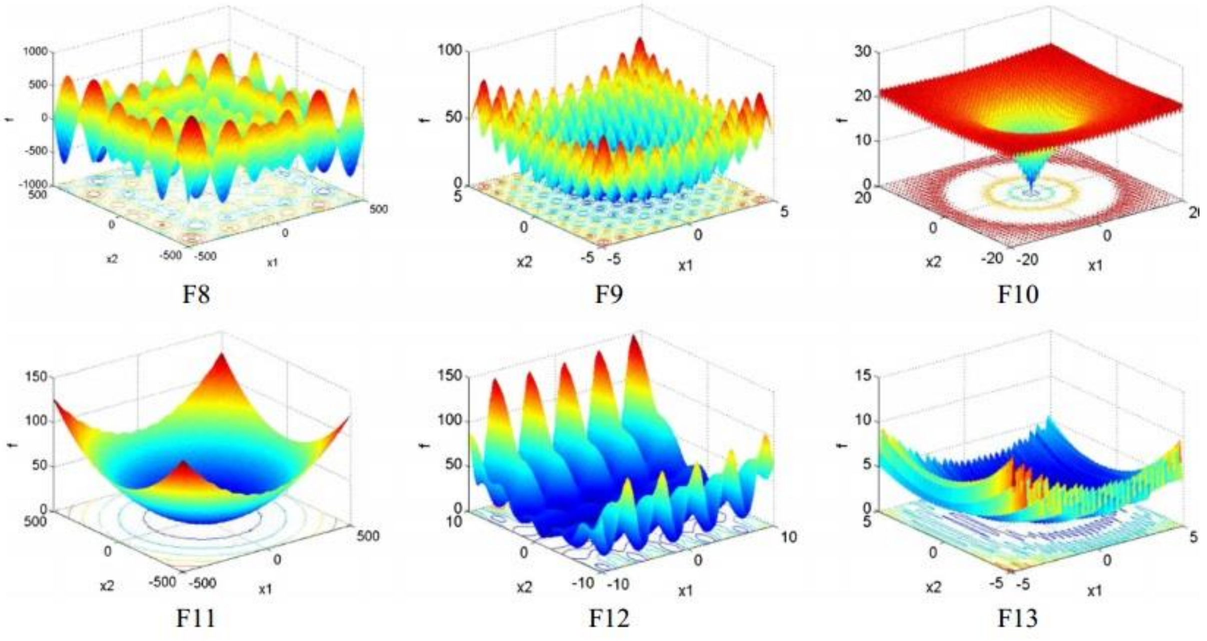

Fig. 3. Search space in multimodal test functions.

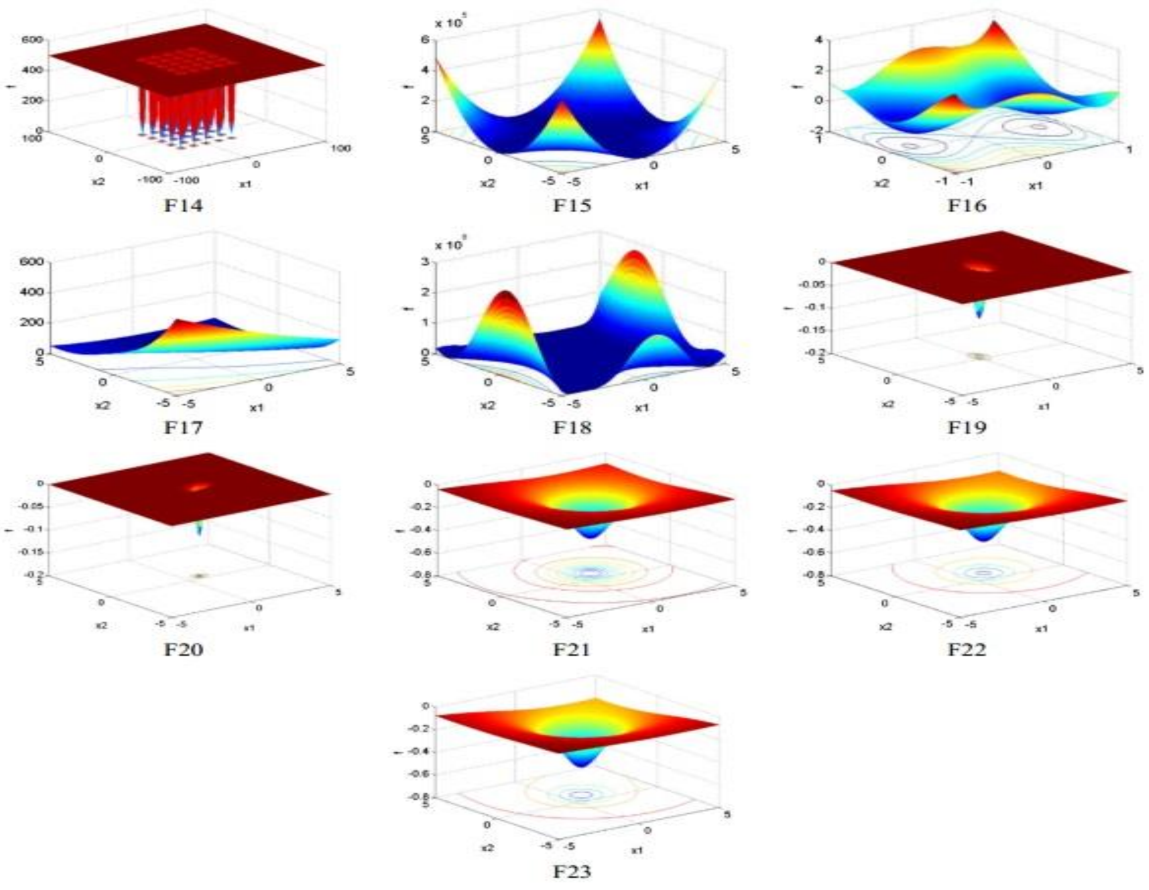

Fig. 4. Search space in fixed-dimension multimodal test functions.

Difference of Evolutionary Strategy Algorithm with Educational System-Based Optimization Algorithm

- In the educational system-based algorithm, the motion operator of the members is used to the best of them, which is not in the evolutionary strategy algorithm.
- An evolutionary strategy, which is a family of evolutionary algorithms, has junction and intersection operators that are inherently different with the operators of the training algorithm.
- The mutant algorithm proposed by the algorithm is different from the evolutionary strategy algorithm.
- In the evolutionary strategy algorithm, the parameters of the algorithm are a part of the chromosome, but not in the proposed algorithm.
- In an evolutionary strategy, a jump operator is used, in which the size of the steps is set as part of the candidate solution, while such an operator does not exist in the proposed algorithm.

Difference of colonial competition algorithm with educational system-based optimization algorithm

- The difference between the proposed algorithm and ICA's colonial competition algorithm is that in the colonial competition algorithm, data is divided into several categories that are called a country, but ultimately these countries are all divided into one country. If the number of batches is fixed in the proposed algorithm.
- The mutation operator in the colonial competition algorithm is completely different from the proposed algorithm.
- Operator moving toward the goal in the colonial competition algorithm is completely different from the proposed algorithm.

Differentiation of TLBO algorithm with training-based optimization algorithm

- The TLBO algorithm is based on a class and a master. While the proposed algorithm is based on an educational system, several classes have master and teacher support.
- In addition to this, the method of mutation in the proposed algorithm completely differs from similar algorithms.
- The TLBO algorithm is different from the proposed algorithm, as the students move towards the difference between students' average
- The performance of students in changing their position in the TLBO algorithm is quite different from that of the proposed algorithm.

The reason for the convergence of the algorithm and finding the global extension

In this algorithm, in each student's class, they move toward the professor and the professor. The professor is the best class member, but the help professor is not the best class member and this does not cause early convergence. Also, having a lazy student as a mutation operator is also due to the cause.

4. CONCLUSION

The method mentioned in this article is a new metaheuristic algorithm for optimization problems, which is based on an educational system. This algorithm showed a better performance than the other algorithms. In future studies, we will intend to offer another model of the algorithm. Also, solving real problems with this algorithm suggested the best performance of this algorithm. This algorithm can improve by changing some parameters such as the number of classes, number of professors, and number of assistant professors.

REFERENCE

[1] Holland JH. **"Genetic Algorithms".** *Sci Am,* pp. 267:66–72, 1992.

[2] Dorigo M, Birattari M, Stutzle T. **"Ant Colony Optimization".** *Comput Intell agaz,IEEE,* pp. 1:28–39, 2006.

[3] S. Kirkpatrick, C. Gelatt, M. Vecchi, **"Optimization by Simulated Annealing",** *Science 220*, pp. 671–680, 1983.

[4] Talbi, El-Ghazali. **"Metaheuristics: From Design to Impelementation",** *John Wiley and sons* 2009.

[5] Ji, M. and Tang, H. **"Global Optimizations and Tabu Search Based on Mamory".** *Applied Mathematics and Computation*. Vol. 159, pp. 449 – 457, 2004.

[6] Eiben, A.E., Smith, J.E., **"Introduction to Evolutionary Computiong",** *Springer* 2003.

[7] I. Boussaid, J. Lepagnot, P. Siarry, **"A Survey on Optimization Metaheuristics",** *Inform. Sci. 237 (2013) 82–117. ISSN 0020 0255,http://dx.doi.org/10.1016/ j.ins.2013.02.041* (10.07.13).

[8] Rechenberg I. **"Evolution Strategy".** *Comput Intel Imitat Life*, 1, 1994.

[9] Yao X, Liu Y, Lin G. **"Evolutionary Programming Made Faster. Evolut Comput",** *IEEE Trans,* Vol.3, pp. 82–102, 1999.

[10] Digalakis J, Margaritis K. **"On Benchmarking Functions for Genetic Algorithms".** *Int J Comput Math,* Vol. 77, pp. 481–506, 2001.

[11] Molga M, Smutnicki C. **"Test Functions for Optimization Needs".** *Test functions for optimization needs*; 2005.

[12] Yang X-S. **"Test Problems in Optimization",** *arXiv, preprint arXiv*:1008. 0549; 2010.

[13] Mirjalili S, Lewis A. **"S-shaped Versus V-shaped Transfer Functions for Binary Particle Swarm Optimization".** *Swarm Evolut Comput*, Vol. 9, pp. 1–14, 2013.

[14] Mirjalili S, Mirjalili SM, Yang X. **“Binary Bat Algorithm”.** *Neural Comput Appl, inpress*, DOI: 10.1007/s00521-013-1525-5.

[15] Liang J, Suganthan P, Deb K. **“Novel Composition Test Functions For Numerical Global Optimization”.** *In: Swarm intelligence symposium, 2005. SIS 2005. Proceedings 2005 IEEE*; pp. 68–75, 2005.

[16] Van den Bergh F, Engelbrecht A. **“A Study of Particle Swarm Optimization Particle Trajectories”.** *Inf Sci*, Vol. 176, pp. 937–71, 2006.

[17] Liang J, Suganthan P, Deb K. **“Novel Composition Test Functions for Numerical Global Optimization”.** *In: Swarm intelligence symposium, 2005. SIS 2005. Proceedings 2005 IEEE*; pp. 68–75, 2005.

[18] Kennedy J, Eberhart R. **“Particle Swarm Optimization”,** *in Neural Networks, 1995.In: Proceedings, IEEE international conference on*; pp. 1942–1948, 1995.

[19] Rashedi E, Nezamabadi-Pour H, Saryazdi S. **“GSA: a Gravitational Search Algorithm”.** *Inf Sci* Vol. 179, pp. 2232–48, 2009.

[20] Storn R, Price K. **“Differential Evolution – A Simple and Efficient Heuristic for Global Optimization Over Continuous Spaces”.** *J Global Optim*, Vol. 11, pp. 341–59, 1997.

[21] Yao X, Liu Y, Lin G. **“Evolutionary Programming Made Faster”.** *Evolut Comput, IEEE Trans*, Vol. 3, pp. 82–102, 1999.

[22] Mirjalili, Seyedali, Seyed Mohammad Mirjalili, and Andrew Lewis. **“Grey Wolf Optimizer”.** *Advances in Engineering Software* 69, pp. 46-61, 2014.

[23] A. Baykasoğlu, A. Hamzadayi, and S. Y. Köse, **"Testing the Performance Of Teaching–Learning Based Optimization (Tlbo) Algorithm on Combinatorial Problems: Flow Shop and Job Shop Scheduling Cases",** ***I**nformation Sciences,* Vol. 276, pp. 204-218, 2014.

[24] Z. Alrashdi and M. Sayyafzade, **"(μ+ λ) Evolution strategy Algorithm in Well Placement, Trajectory, Control and Joint Optimisation,"** *Journal of Petroleum Science and Engineering*, 2019.

[25] A. Mollajan, H. Memarian, and B. Quintal, **"Imperialist Competitive Algorithm (ICA) Optimization Method for NonlineaR AVA Inversion,"** *Geophysics*, Vol. 84, No. 3, pp. 1-72, 2019.