[1.]Ismail KOC, [2.]Huseyin HAKLI

A NEW MUTATION OPERATOR FOR DIFFERENTIAL EVOLUTION ALGORITHM IN SOLVING CONTINUOUS PROBLEMS

[1.]Selcuk University, Engineering Faculty, Computer Engineering Department, Konya, TURKEY
[2.]Necmettin Erbakan University, Engineering & Architecture Faculty, Computer Engineering Department, Konya, TURKEY

Abstract: Differential evolution (DE) is one of the most popular and powerful evolutionary algorithms for numerical optimization. DE is a simple but very effective population based search technique. In this study, a new mutation operator has been proposed to generate new individuals (offspring) from elitist individuals. Two crossover operators, binomial and exponential, are used as the crossover operator. The proposed mutation operator is compared with five different DE mutation operators used frequently in the literature using binomial and exponential crossover operators: DE/rand/1, DE/best/1, DE/current_to_best/1, DE/best/2 and DE/rand/2. In the experimental studies carried out using 17 different benchmarks problems, it has been observed that the proposed mutation operator is the best method after DE/rand/1 and DE/best/2 mutation operators especially for binomial crossover operator. These results show that the proposed mutation operator can be used as an alternative for solving continuous optimization problems.
Keywords: continuous optimization, algorithms for numerical optimization, differential evolution (DE)

1. INTRODUCTION

Nature-inspired algorithms have been proposed to solve optimization problems which have different characteristics (continuous, discrete, constrained etc.) within the reasonable time. These algorithms can be separated two groups: swarm intelligence and evolutionary algorithms [1]. Swarm intelligence-based algorithms such as particle swarm optimization [2], artificial bee colony [3], ant colony optimization [4] use a solution search equation to generate the new individual. Instead of using a solution search equation, evolutionary algorithms use mutation and crossover operators for generating the new individual. The most known evolutionary algorithms in the literature are genetic algorithm [4], differential evolution algorithms (DE) [5] and scatter search [6].

The DE algorithm, proposed by Storn and Price [5], and it was implemented to many real-world optimization problems such as baker's yeast drying process [7], energy demand estimation [8], image thresholding [9], raw milk transportation [10], multiple container loading problems [11] by virtue of easy adaption, powerful performance and less parameters.

The mutation and crossover operators are important processes of DE algorithm and, there are several schemes/strategies for these operators [12]. Due to the fact that the strategies of these operators directly affect the performance of DE algorithm, many studies have been performed about this subject in the literature to improve and to enhance the performance of DE algorithm. A self-adaptive DE algorithm with discrete mutation control parameters (DMPSADE) is proposed by Fan and Yan [13] to balance the exploitation and exploration. Zhou et al. proposed a novel differential evolution (DE) algorithm with intersect mutation operation [14]. In their study, the population is divided into the better part and the worse part in accordance with the fitness value and then novel mutation and crossover operators are used for generating the new individuals. In another study, four popular mutation operators, “rand/1,” “rand/2,” “best/1,” and “best/2”, are employed adaptively to select the target individual in the population [15]. This proposed mutation scheme provides the balance between local and global search and maintains local exploitation abilities for DE algorithm. Asafuddoula et al. proposed an adaptive hybrid DE algorithm and this algorithm performs a binomial crossover in early stages of evolution and then exponential crossover in later stages [16]. A new triangular mutation rule was presented for mutation operator in another study; it was based on the convex combination vector of the triangle and the difference vector between the best and the worst individuals among the three randomly selected vectors [17]. This approach has shown a better performance than the basic DE in accordance with global and local search capabilities and convergence speed. Zou et al. proposed the improved DE (IDE) which has three modifications: 1) two mutation operators and 2) a dynamical crossover rate are used, and 3) a useful population randomization is adopted [18]. To dynamically tune the mutation factor of DE and improve its exploration and exploitation, a new approach of differential evolution (DE) which uses fuzzy logic inference system was proposed by Salehpour et al [19]. Besides the improvements of DE algorithm, DE was hybridized with other nature-inspired algorithms such as artificial bee colony [20], teaching-learning based optimization [21], harmony search [22], particle swarm optimization [23].

The impact of the mutation and crossover operators on the DE algorithm is huge as seen in studies about improvements on the DE in the literature. In addition, DE has a powerful ability on the exploration due to randomness on the mutation operators [24]. Therefore, a new mutation operator based on the elitist strategy is proposed to enhance its exploitation ability in this study. In this strategy, besides the three random individuals, two elite individuals in the population are selected randomly from the elite individuals to be used in mutation operator. To investigate and analysis the performance of this new mutation operator, the proposed mutation operator is compared with five different mutation operators (DE/rand/1, DE/rand/2, DE/best/1, DE/best/2, DE/rand-to-best/1) on the 17 benchmark functions. Moreover, crossover process is performed with two different methods (binomial and exponential) separately in the experiments. Experimental studies carried out using two different crossover operators (binomial and exponential) show that the proposed method achieves reasonable results at a competitive level especially for binomial crossover.

The remainder of this paper is divided as follows. The Section 2 explains the basic DE and Section 3 describes the proposed mutation operator. Then, experimental results are reported and evaluated in Section 4. Finally the paper is concluded in the last section.

2. DIFFERENTIAL EVOLUTION ALGORITHM

Differential Evolution Algorithm (DE) is a population-based optimization technique proposed by Price and Storn in 1995, which is similar to the genetic algorithm in terms of operations and operators. The DE algorithm is able to produce effective results, especially in continuous problems [25]. DE which is able to conduct research at many points in the search space at the same time investigates better results for the solution of the problem with the help of its operators through the generations. Although similar to GA, unlike classical binary GA, variables in DE method are represented by their real values [26]. The crossover, mutation, and selection operators in GA are used in the DE method, but their usage forms and sequences differ from each other. In DE, individuals are handled one by one, and a new individual is obtained using the other three individuals selected randomly. Mutation and crossover operators are used to perform these operations [27]. The flowchart of DE is presented in Figure 1. The operators in the DE algorithm are briefly as follows:

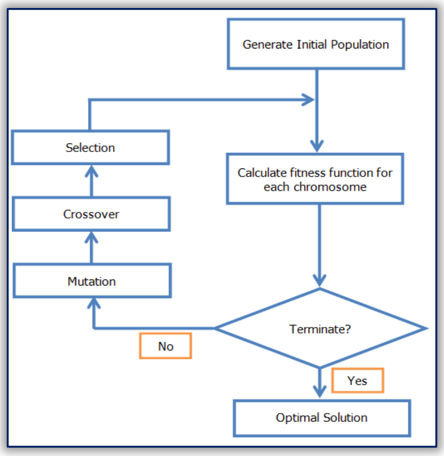

Figure 1. Flowchart of DE

— Mutation

The mutation operator is the main operator that allows DE to be different from other evolutionary algorithms. The goal of the mutation is to make changes at random rates on some genes of the current individual. Thanks to these changes, the solution point represented by the individual moves at a certain distance in the solution space. In order to achieve the goal of this process, it is necessary to determine the changes that will provide the correct direction and amount of movement. The mutation process in each generation starts by randomly selecting three different individuals in the population. The most frequently used mutation strategies applied in DE codes are given in Eq. (1–5).

$$\text{DE/rand/1:}\quad \mathbf{V_{i,g}} = \mathbf{X_{r1,g}} + \mathbf{F} * (\mathbf{X_{r2,g}} - \mathbf{X_{r3,g}}) \tag{1}$$

$$\text{DE/rand/2:}\quad V_{i,g} = X_{r1,g} + F * (X_{r2,g} - X_{r3,g}) + F * (X_{r4,g} - X_{r5,g}) \tag{2}$$

$$\text{DE/best/1:}\quad V_{i,g} = X_{best,g} + F * (X_{r1,g} - X_{r2,g}) \tag{3}$$

$$\text{DE/best/2:}\quad V_{i,g} = X_{best,g} + F * \left(X_{r1,g} - X_{r2,g}\right) + F * \left(X_{r3,g} - X_{r4,g}\right) \tag{4}$$

$$\text{DE/rand-to-best/1:}\quad V_{i,g} = X_{x1,g} + F * \left(X_{best,g} - X_{r2,g}\right) + F * \left(X_{r3,g} - X_{r4,g}\right) \tag{5}$$

NP represents the population number, i = 1,. . . , NP, r1, r2, r3 ∈ [1,. . . , NP] are selected randomly. Furthermore, r1 ≠ r2 ≠ r3 ≠ i, and F ∈ [0, 1] is a mutation scale parameter that is preferred.

— Crossover

When crossover process is performed, the candidate individual ($U_{i,G+1}$) is generated for the new generation using the difference individual ($V_{i,G}$) obtained from the mutation and the current individual ($X_{i,G}$). When the candidate trial individual will be generated, each gene in the candidate individual is taken from the difference individual with a probability of CR, and it is taken from the current individual with a probability of 1- CR. The crossover operator is given in Eq. (6).

$$U_{j,i,G+1} = \begin{cases} V_{j,i,G+1} & \text{if } rand_j \le CR \vee j = k \\ X_{j,i,G+1} & \text{otherwise} \end{cases} \tag{6}$$

In Eq. (6), j = 1.. n, k ∈ [1,. . . , n] is the random parameter index chosen once for each *i*, and this value is determined by the user, with the control parameter CR ∈ [0, 1].

Exponential crossover is another commonly used crossover operator. In this crossover type, genes values are copied to the trial vector $U_{i(t)}$ from the mutant vector $V_{i(t)}$ starting at a randomly chosen position. This process continues until the condition randj[0,1] > CR is met. The rest of the genes are taken from the target vector $X_{i(t)}$.

— Selection

The selection scheme of DE differs from other evolutionary algorithms. For the next generation, the population is selected from the individual in the current population according to the current rule and from the respective trial vector. The selection operator is given in Eq. (7).

$$X_{i,G+1} = \begin{cases} U_{i,G+1} & \text{if } f(U_{i,G+1}) \leq f(X_{i,G}) \\ X_{i,G} & \text{otherwise} \end{cases} \quad (7)$$

3. PROPOSED MUTATION OPERATOR

Mutation operator is the fundamental operator that increases the ability to find the optimal solution in search space. Until now, very different mutation operators have been proposed in the literature. When the mutation operators in the literature are examined, the best individual-based or completely random-based approaches are proposed. In this study, an elitism based approach is proposed. According to this approach, two elite individuals randomly selected from among the elite (best) individuals in the population have been used instead of using just the best individual. The proposed mutation operator is given in Eq. (8).

$$V_{i,g} = X_{r1,g} + F * (X_{elitism1,g} - X_{r2,g}) + F * (X_{elitism2,g} - X_{r3,g}) \quad (8)$$

In Eq. (8), *elitism1* and *elitism2* are random indices selected randomly from the elite individuals and *elitism1* and *elitism2* must be different from each other. In addition, in this study, rate of elite individuals was determined as 0.1.

4. EXPERIMENTAL STUDIES

In order to compare the proposed mutation operator with the other operators, seventeen different benchmark functions with different characteristics were used in experimental studies. These benchmark functions are given in Table 1 in detail.

Table 1. Benchmark functions

	Range	C	Function	Formulation
F1	[-5.12, 5.12]	U	Sphere	$f_1 = \sum_{i=1}^{n} x_i^2$
F2	[-10, 10]	U	Schwefel 2.22	$f_2 = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$
F3	[-10, 10]	U	Rosenbrock	$f_3 = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$
F4	[-1.28, 1.28]	U	Noise	$f_4 = \sum_{i=1}^{n} ix_i^4 + random[0,1]$
F5	[-500, 500]	M	Schwefel	$f_5 = 418.98288727243369 * n - \sum_{i=1}^{n} x_i \sin(\sqrt{\lvert x_i \rvert})$
F6	[-5.12, 5.12]	M	Rastrigin	$f_6 = \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_{i)+10}]$
F7	[-32, 32]	M	Ackley	$f_7 = -20\exp\left\{-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right\} - exp\left\{\frac{1}{n}\sum_{i=1}^{n} \cos(2\pi x_i)\right\} + 20 + e$
F8	[-600, 600]	M	Griewank	$f_8 = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$
F9	[-10, 10]	U	SumSquare	$f_9 = \sum_{i=1}^{n} ix_i^2$
F10	[-100, 100]	U	Step	$f_{10} = \sum_{i=1}^{n} (\lfloor x_i + 0.5 \rfloor)^2$
F11	[-1.28, 1.28]	U	Quartic	$f_{11} = \sum_{i=1}^{n} ix_i^4$
F12	[-10, 10]	M	Levy	$f_{12} = \sum_{i=1}^{n-1} (x_i - 1)^2 [1 + sin^2(3\pi x_{i+1}] + sin^2(3\pi x_1) + \lvert x_n - 1 \rvert [1 + sin^2(3\pi 3_n)]$

F13	[-100, 100]	M	Schaffer	$f_{13} = 0.5 + \frac{sin^2\left(\sqrt{\sum_{i=1}^{n} x_i^2}\right) - 0.5}{\left(1 + 0.001\left(\sum_{i=1}^{n} x_i^2\right)\right)^2}$
F14	[-10, 10]	M	Alpine	$f_{14} = \sum_{i=1}^{n} \lvert x_i sin(x_i) + 0.1x_i \rvert$
F15	[-5.12, 5.12]	M	Non-Continuos Rastrigin	$f_{15} = \sum_{i=1}^{n} [y_i^2 - 10cos(2\pi y_i) + 10] \quad y_i = \begin{cases} x_i, & \lvert x_i \rvert < \frac{1}{2} \\ \frac{round(2x_i)}{2}, & \lvert x_i \rvert \geq \frac{1}{2} \end{cases}$
F16	[-5, 10]	U	Zakharov	$f_{16} = \sum_{i=1}^{n} x_i^2 + \left(\sum_{i=1}^{n} 0.5ix_i\right)^2 + \left(\sum_{i=1}^{n} 0.5ix_i\right)^4$
F17	[-100, 100]	U	Elliptic	$f_{17} = \sum_{i=1}^{n} (10^6)^{(i-1)/(n-1)} x_i^2$

The proposed operator was compared with the five mutation operators most commonly used in the literature. Furthermore, two different general comparisons were carried out separately using two different crossover operators namely binomial and exponential crossover methods. In all experimental studies, the parameter values were taken equal in order to perform comparisons fairly. The parameter values of DE are given in Table 2.

Table 2. Parameter values of DE

Parameters	Values
Population size	50
CR	0.9
F	0.5

Table 3. Comparative experimental results for binomial crossover

Functions	Rand1	Best1	Current Best1	Best2	Rand2	Proposed Method
F1	2.35E-44	2.72E+00	6.30E-01	5.46E-77	3.19E-01	5.51E-24
	2	6	5	1	4	3
F2	3.87E+01	5.68E+00	5.49E-01	1.39E+01	6.56E+01	6.10E+00
	4	2	1	3	6	5
F3	1.42E+01	5.12E+02	3.47E+02	1.20E+00	5.58E+00	3.99E-01
	4	6	5	2	3	1
F4	2.87E-01	1.31E-01	1.20E-01	3.58E-01	1.48E-01	6.29E-01
	1	5	4	2	6	3
F5	1.20E+01	2.88E+02	2.38E+02	1.85E+02	4.16E+02	2.33E+02
	1	5	4	2	6	3
F6	2.81E+01	5.54E+01	2.14E+01	9.51E+01	1.19E+02	8.24E+01
	2	3	1	5	6	4
F7	2.84E-01	8.44E+00	4.41E+00	3.20E-01	2.40E-01	3.34E+00
	1	6	5	3	2	4
F8	1.85E-01	1.60E+00	2.82E+00	1.53E-01	3.59E-01	1.00E-01
	1	6	5	3	4	2
F9	1.50E-42	1.20E+01	3.14E+01	3.61E-75	6.17E-01	1.43E-22
	2	6	5	1	4	3
F10	0.00E+00	8.85E+02	2.91E+02	1.66E-32	9.33E-01	1.72E-21
	1	6	5	2	4	3
F11	2.87E-02	8.78E-01	5.41E-01	2.59E-128	1.47E-01	3.00E-41
	3	6	5	1	4	2
F12	1.50E-32	5.46E+00	1.64E+00	1.17E+00	1.67E-01	6.81E-01
	1	6	5	4	2	3
F13	2.22E-01	3.95E-01	2.48E-01	6.46E-01	6.90E-01	3.21E-01
	1	6	5	3	4	2
F14	4.92E-41	1.98E+00	3.70E-01	3.50E-01	5.66E+00	2.42E-01
	1	5	4	3	6	2
F15	2.77E+01	5.17E+01	2.43E+01	7.28E+01	1.20E+01	7.20E+01
	2	3	1	5	6	4
F16	5.57E-01	2.00E+00	1.29E+01	8.70E-03	4.35E-01	3.21E-01
	2	6	5	1	4	3
F17	1.95E-39	2.93E+02	7.59E+02	2.38E-07	4.88E-01	1.57E+02
	2	6	5	1	3	4
Rank Values	1.82	5.24	4.12	2.48	4.35	3.00
Final Rank	1	6	4	2	5	3

The average values obtained by the different mutation operators for each function are given in Table 3 and Table 4. According to mean values, the proposed mutation operator was compared with other methods in the literature. Considering these results, rank values were determined for each function and finally, final rank values were obtained according to these values. When the final rank values are examined, it is seen that the proposed method ranks 3rd after Rand1 and Best2 operators. It is also seen that the proposed method has a near rank value to the Best2 method. These results show that the proposed mutation method produced good results at a reasonable level. When Table 4 is examined, a different result was obtained than the results in Table 3. It is observed that the method proposed in Table 3 is the most successful result after Rand1 and Best2 methods whereas in Table 4 it is clearly seen to be more successful than only Best1 and Best2 methods. As a result, it can be said that the proposed method is more successful especially for binomial crossover.

Table 4. Comparative experimental results for exponential crossover

Functions	Rand1	Best1	Current_Best1	Best2	Rand2	Proposed Method
F1	3.25E+00	1.26E+01	4.32E+00	4.83E+00	7.73E-01	3.95E+00
	2	6	4	5	1	3
F2	3.62E+01	3.35E+01	7.35E+00	1.66E+01	6.71E+01	6.47E+01
	4	3	1	2	6	5
F3	3.33E+03	1.54E+04	2.41E+03	4.60E+03	2.21E+02	3.19E+03
	4	6	2	5	1	3
F4	3.85E-02	8.23E-01	1.51E-01	1.41E-01	1.70E-02	1.56E-01
	2	6	4	3	1	5
F5	3.36E+03	4.07E+03	3.43E+03	2.80E+03	3.50E+03	2.74E+03
	3	6	4	2	5	1
F6	7.72E+01	1.19E+02	5.25E+01	6.02E+01	1.11E+02	6.78E+01
	4	6	1	2	5	3
F7	8.82E+00	1.33E+01	8.65E+00	1.01E+01	2.62E+00	1.13E+01
	3	6	2	4	1	5
F8	1.39E+01	4.44E+01	1.62E+01	1.76E+01	3.26E+00	2.11E+01
	2	6	3	4	1	5
F9	1.48E+02	4.57E+02	1.27E+02	1.68E+02	1.87E+01	1.34E+02
	4	6	2	5	1	3
F10	1.63E+03	4.20E+03	1.24E+03	1.84E+03	2.47E+02	2.37E+03
	3	6	2	4	1	5
F11	5.26E-02	6.95E-01	6.09E-02	7.68E-02	1.76E-03	8.04E-02
	2	6	3	4	1	5
F12	2.61E+00	1.14E+01	3.00E+00	4.12E+00	6.21E-01	3.45E+00
	2	6	3	5	1	4
F13	4.30E-01	4.86E-01	4.24E-01	4.71E-01	2.61E-01	4.66E-01
	3	6	2	5	1	4
F14	2.67E+00	8.42E+00	4.00E+00	3.35E+00	4.84E-01	2.11E+00
	3	6	5	4	1	2
F15	6.27E+01	8.04E+01	3.30E+01	6.04E+01	9.33E+01	7.09E+01
	3	5	1	2	6	4
F16	3.68E+01	1.72E+02	4.64E+01	5.70E+01	5.48E+00	3.71E+01
	2	6	4	5	1	3
F17	1.04E+05	1.41E+06	1.88E+05	1.43E+05	4.45E+03	1.23E+05
	2	6	5	4	1	3
Rank Values	2.82	5.76	2.82	3.82	2.06	3.71
Final Rank	2	6	2	5	1	4

5. CONCLUSIONS

In this study, we proposed a new mutation operator for the DE algorithm, one of the metaheuristic optimization algorithms. Instead of using only the best individual in some mutation operators in the literature, we proposed a new elitist-based mutation operator considering two randomly selected individuals among the best individuals in the population. In order to comparatively analyze the results, five different mutation methods used most frequently in the literature have been used in experimental studies. In addition, two different crossover operators for experimental studies were used with seventeen different benchmark functions in two different experiment sets and these sets were run under exactly the same conditions in order to compare each other fairly. It has been clearly seen that the proposed mutation operator is the best method after DE/rand/1 and DE/best/2 mutation operators especially for binomial crossover operator. Therefore, the proposed mutation operator may be used as an alternative for solving other continuous optimization problems.

Note

This paper is based on the paper presented at INTERNATIONAL CONFERENCE ON APPLIED SCIENCES – ICAS 2018, organized by UNIVERSITY POLITEHNICA TIMISOARA, Faculty of Engineering Hunedoara (ROMANIA) and UNIVERSITY OF BANJA LUKA, Faculty of Mechanical Engineering (BOSNIA & HERZEGOVINA), in cooperation with the Academy of Romanian Scientists, Academy of Sciences Republic of Srpska, Academy of Technical Sciences of Romania – Timisoara Branch and General Association of Romanian Engineers – Hunedoara Branch, in Banja Luka, BOSNIA & HERZEGOVINA, 9 – 11 May 2018.

References

[1] A. Banitalebi, M. I. A. Aziz, A. Bahar, and Z. A. Aziz, "Enhanced compact artificial bee colony," *Information Sciences,* vol. 298, pp. 491-511, 2015.
[2] J. Kennedy, "Particle swarm optimization," in *Encyclopedia of machine learning*, ed: Springer, 2011, pp. 760-766.
[3] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm," *Journal of global optimization,* vol. 39, pp. 459-471, 2007.
[4] N. M. Al Salami, "Ant colony optimization algorithm," *UbiCC Journal,* vol. 4, pp. 823-826, 2009.
[5] R. Storn and K. Price, "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces," *Journal of global optimization,* vol. 11, pp. 341-359, 1997.

[6] F. Glover, M. Laguna, and R. Martí, "Fundamentals of scatter search and path relinking," *Control and cybernetics,* vol. 29, pp. 653-684, 2000.
[7] U. Yüzgeç and M. Eser, "Chaotic based differential evolution algorithm for optimization of baker's yeast drying process," *Egyptian Informatics Journal,* 2018.
[8] M. Beskirli, H. Hakli, and H. Kodaz, "The energy demand estimation for Turkey using differential evolution algorithm," *Sādhanā,* vol. 42, pp. 1705-1715, 2017.
[9] N. Muangkote, K. Sunat, and S. Chiewchanwattana, "Rr-cr-IJADE: An efficient differential evolution algorithm for multilevel image thresholding," *Expert Systems with Applications,* vol. 90, pp. 272-289, 2017.
[10] K. Sethanan and R. Pitakaso, "Differential evolution algorithms for scheduling raw milk transportation," *Computers and Electronics in Agriculture,* vol. 121, pp. 245-259, 2016.
[11] X. Li and K. Zhang, "A hybrid differential evolution algorithm for multiple container loading problem with heterogeneous containers," *Computers & Industrial Engineering,* vol. 90, pp. 305-313, 2015.
[12] M. Z. Ali, N. H. Awad, and P. N. Suganthan, "Multi-population differential evolution with balanced ensemble of mutation strategies for large-scale global optimization," *Applied Soft Computing,* vol. 33, pp. 304-327, 2015.
[13] Q. Fan and X. Yan, "Self-adaptive differential evolution algorithm with discrete mutation control parameters," *Expert Systems with Applications,* vol. 42, pp. 1551-1572, 2015.
[14] Y. Zhou, X. Li, and L. Gao, "A differential evolution algorithm with intersect mutation operator," *Applied Soft Computing,* vol. 13, pp. 390-401, 2013.
[15] V. Ho-Huu, T. Vo-Duy, T. Luu-Van, L. Le-Anh, and T. Nguyen-Thoi, "Optimal design of truss structures with frequency constraints using improved differential evolution algorithm based on an adaptive mutation scheme," *Automation in Construction,* vol. 68, pp. 81-94, 2016.
[16] M. Asafuddoula, T. Ray, and R. Sarker, "An adaptive hybrid differential evolution algorithm for single objective optimization," *Applied Mathematics and Computation,* vol. 231, pp. 601-618, 2014.
[17] A. W. Mohamed, "Solving stochastic programming problems using new approach to Differential Evolution algorithm," *Egyptian Informatics Journal,* vol. 18, pp. 75-86, 2017.
[18] D. Zou, S. Li, G.-G. Wang, Z. Li, and H. Ouyang, "An improved differential evolution algorithm for the economic load dispatch problems with or without valve-point effects," *Applied Energy,* vol. 181, pp. 375-390, 2016.
[19] M. Salehpour, A. Jamali, A. Bagheri, and N. Nariman-zadeh, "A new adaptive differential evolution optimization algorithm based on fuzzy inference system," *Engineering Science and Technology, an International Journal,* vol. 20, pp. 587-597, 2017.
[20] S. S. Jadon, R. Tiwari, H. Sharma, and J. C. Bansal, "Hybrid artificial bee colony algorithm with differential evolution," *Applied Soft Computing,* vol. 58, pp. 11-24, 2017.
[21] O. E. Turgut and M. T. Coban, "Optimal proton exchange membrane fuel cell modelling based on hybrid Teaching Learning Based Optimization–Differential Evolution algorithm," *Ain Shams Engineering Journal,* vol. 7, pp. 347-360, 2016.
[22] J. Zhang, Y. Wu, Y. Guo, B. Wang, H. Wang, and H. Liu, "A hybrid harmony search algorithm with differential evolution for day-ahead scheduling problem of a microgrid with consideration of power flow constraints," *Applied energy,* vol. 183, pp. 791-804, 2016.
[23] S. Sayah and A. Hamouda, "A hybrid differential evolution algorithm based on particle swarm optimization for nonconvex economic dispatch problems," *Applied Soft Computing,* vol. 13, pp. 1608-1619, 2013.
[24] J.-T. Tsai, "Improved differential evolution algorithm for nonlinear programming and engineering design problems," *Neurocomputing,* vol. 148, pp. 628-640, 2015.
[25] D. Mayer, B. Kinghorn, and A. Archer, "Differential evolution–an easy and efficient evolutionary algorithm for model optimisation," *Agricultural Systems,* vol. 83, pp. 315-328, 2005.
[26] O. Hrstka and A. Kučerová, "Improvements of real coded genetic algorithms based on differential operators preventing premature convergence," *Advances in Engineering Software,* vol. 35, pp. 237-246, 2004.
[27] T. Keskintürk, "Diferansiyel gelişim algoritması," *İstanbul Ticaret Üniversitesi Fen Bilimleri Dergisi,* vol. 5, pp. 85-99, 2006.

ISSN 1584 - 2665 (printed version); ISSN 2601 - 2332 (online); ISSN-L 1584 - 2665