Clustering based Adaptive Differential Evolution for Numerical Optimization

Bilal
Dept. of Applied Science & Engineering
Indian Institue of Technology Roorkee
Roorkee, India
bilal25iitr@gmail.com

Millie Pant
Dept. of Applied Science & Engineering
Indian Institue of Technology Roorkee
Roorkee, India
millifpt@iitr.ac.in

Garima Vig
Dept. of Computer Science Engineering
College of Engineering Roorkee
Roorkee, India
garimavig311@gmail.com

Abstract— **In this paper, a fuzzy C-means clustering approach is suggested for segregating the initial population of Differential Evolution (DE) on the basis of the membership function. The proposed algorithm called FCADE further incorporates adaptive crossover and mutation strategies into the segregated population. Three variants of FCADE are proposed and are applied to selected CEC2005 benchmark problems. The results, when compared with some of the well-known adaptive algorithms indicates the competence of the strategies proposed in the present study in enhancing the performance of the DE algorithm.**

Keywords—Differential Evolution, Fuzzy Means Clustering, Mutation, Crossover, Adaptive.

I. INTRODUCTION

Differential evolution (DE), first proposed by Storn and Price[1] at Berkeley, is a simple yet powerful evolutionary algorithm [2]–[10]. The DE family of algorithms has been frequently adopted to tackle multi-objective, constrained, dynamic, large-scale, and multimodal optimization problems. DE variants have been achieving top ranks in various competitions held under the IEEE Congress on Evolutionary Computation (CEC) conference series (e.g., see https://www3.ntu.edu.sg/home/epnsugan/index_files/).

One of the trends that attracted the attention of researchers working in DE is embedding adaptive behavior into the DE population and also on mutation and crossover strategies.

The concept of the adaptive population was first suggested by Teo in ([22], [23]). Deng et al. [24] developed an improved self-adaptive differential evolution algorithm with multiple strategies (ISDEMS). In ISDEMS, the population is dynamically divided into multiple sub populations according to the fitness value of the individuals. Multiple strategies are used to improve the diversity of the individuals, to avoid premature convergence and to ensure efficiency in exchanging information among subpopulations. In[25], authors have proposed a DE variant called adaptive population topology differential evolution algorithm (APTDE) for solving unconstrained problems. The authors showed that their method helps in avoiding premature convergence by updating the population topology adaptively. Aalto and Lampinen [26] proposed an adaptive mechanism population-based DE called cumu-DE in which a probability mass function mechanism is used for automatically adapting the population size.

Selected references on adaptive crossover and mutation strategies are as follows: Islam et al. [27] proposed a new mutation strategy with binomial crossover and two adaptive control parameters. In their work, a biased selection method is introduced where the mutant vector undergoes crossover with 'p' top-ranked population vectors, rather than with a single target vector. The authors validated the results on 25 CEC 2005 benchmark problems. Qin and Suganthan [4] proposed a self-adaptive DE (SaDE). In SaDE, suitable learning strategy and parameter settings are gradually self-adapted according to the learning experience during the evolution. Brest et. al [28] proposed an adaptive DE called jDE and tested its applicability on benchmark test problems. Mallipeddi et. al [29] developed a DE named EPSDE with an ensemble of parameters and mutation and crossover strategies. Zhang [30] proposed adaptive DE (JADE) with an optional external archive and validated its performance on benchmark test problems. Liu and Lampinen [**41**] proposed FADE, in which they used fuzzy logic to develop adaptive control parameters *F* and *Cr*.

DE variant, FCADE, proposed in the present study has two main features: Firstly, it follows the distribution of population points on the basis of Fuzzy C-Means Clustering and secondly, the operators crossover and mutation follows an adaptive strategy. Three variants of FCADE

The rest of the paper is organized as follows: Section II discussed the DE algorithm. The Proposed algorithm is illustrated in Section III. Section IV presents experimental results. Finally, conclusions are given in Section V.

II. DIFFERENTIAL EVOLUTION

DE works in two phases, the first phase is initialization and the second phase is evolution. In the first phase, the population is generated randomly and in the second phase, which is evolution, the generated population goes through mutation, crossover, and selection process which are repeated until termination criteria are met. The main steps of DE are as follows:

a) *Initialization*: During initialization, a set of the uniformly distributed population is generated as follows:

Let $S^G = \{X_j^G : j = 1,2, \dots, NP\}$ be the population at any generation G, NP denotes the size of population. Here, P_j^G denotes a D-dimensional vector as $X_j^G =$

978-1-7281-6929-3/20/$31.00 ©2020 IEEE

$\{x_{1,j}^G, x_{2,j}^G, \ldots, x_{D,j}^G\}$. X_j^G is generated using uniformly distributed random number

$$X_j^G = X_{low} + (X_{upp} - X_{low}) * rand(0,1)$$

b) *Evolution:* This is the second phase where mutation, crossover, and selection operations are performed.

Mutation: During mutation, a mutant vector V_j^G is generated for each target vector X_j^G at generation G as

$$V_j^G = X_{r1}^G + F * (X_{r2}^G - X_{r3}^G)$$

Two commonly used mutation strategies of DE are:

DE/rand/1/bin

$$(V = X_{rand1\_pos} + F * (X_{rand2\_pos} - X_{rand3\_pos}))$$

DE/current-to-best/1/bin

$$(V = X_{best\_pos} + F * (X_{rand2\_pos} - X_{rand3\_pos}))$$

Where F is the scaling factor and the value of F ranges between 0 to 1; $r1, r2, r3 \in \{1,2,\ldots,NP\}$ are randomly chosen vectors different from each other and also from the target vector.

Crossover: After mutation, crossover is performed to generate a new vector called the trial vector denoted as $T_j^G = \{t_{1,j}^G, t_{2,j}^G, \ldots, t_{D,j}^G\}$. Crossover is performed between the target vector $X_j^G = \{x_{1,j}^G, x_{2,j}^G, \ldots, x_{D,j}^G\}$ and mutant vector $V_j^G = \{v_{1,j}^G, v_{2,j}^G, \ldots, v_{D,j}^G\}$ using a crossover probability Cr whose value is between $0\ to\ 1$. T_j^G is generated as

$$t_{i,j}^G = \begin{cases} v_{i,j}^G & if\ rand_j \le Cr \\ x_{i,j}^G & otherwise \end{cases}$$

Where $i \in \{1,2,\ldots,D\}\ and\ Cr \in [0,1]$.

Selection: During this operation, a comparison is done between the target vector and trial vector according to their fitness value. The one having better fitness survives to the next generation. This operation is performed as:

$$X_j^{G+1} = \begin{cases} T_j^G & if\ f(T_j^G) \le f(X_j^G) \\ X_j^G & otherwise \end{cases}$$

These three operations are repeated till a predefined termination criterion is satisfied

III. PROPOSED ALGORITHM

Fuzzy C-means Algorithm: Fuzzy c-means is a special type of clustering algorithm which is based on the fact that a particular data point may belong to more than one group or cluster. It was proposed by Dunn [37] in 1973 and was later improved by Bezdek, J. C [38] in 1980.

Fuzzy c means (FCM) algorithm works by assigning membership to each individual in the population corresponding to each cluster center on the basis of the distance between the cluster center and the data point. Nearer the data to the cluster center, higher will its membership value for that cluster. Clearly, the summation of membership for each data point should be equal to one. Every individual in the population is bound to a cluster by means of a membership function, representing the fuzzy behavior of the algorithm.

Fuzzy c-means algorithm steps:

1. Let $P = \{p_1, p_2, \ldots p_n\}$ be the data set containing n individuals, present in the search space.
2. Fix c (number of clusters), where $2 \le c < n$.
3. Randomly initialize cluster center for c clusters fixed in step 2 represented as $Y = \{y_1, y_2, \ldots, y_c\}$. Create an initial membership matrix U_0 (where '$U = (\mu_{ij})_{n*c}$' is the fuzzy membership matrix for iteration 0) using a pseudo-random initialization strategy.
4. Calculate the fuzzy membership μ_{ki} using formula:

$$\mu_{ki} = \frac{1}{\sum_{j=1}^{c} (d_{ki}/d_{kj})^{(2/(m-1))}} \quad (1)$$

where,

- $d_{ki} = \|p_i - y_k\|$
- $d_{kj} = \|p_i - y_j\|$
- m denotes the level of cluster fuzziness having a value between 0 and ∞.

5. Calculate the fuzzy center using the following formula:

$$y_k = \frac{\sum_{i=1}^{n} \mu_{ki}^m p_i}{\sum_{i=1}^{n} \mu_{ki}^m},\ \forall i = 1,2,3 \ldots\ldots\ldots c \quad (2)$$

where,

- y_k represents k[th] cluster center, and k: [1...c]

6. Calculate Objective function J(X, V) :

$$J(P,Y) = \sum_{i=1}^{n} \sum_{j=1}^{c} \mu_{ij}^m \|p_i - y_j\|^2 \quad (3)$$

Where,

$'||p_i - y_j||'$ is the Euclidean distance between i^{th} data and j^{th} cluster center.

7. Repeat steps 4-6, until $\|U^{itr+1} - U^{itr}\| < \beta$ ('β' is the termination criterion fixed to $1*e^{-8}$) or when the minimum value of J (weighted Sum of Squared Error) is achieved ($1*e^{-8}$).

Fuzzy C-means Adaptive Differential Evolution: The main steps of FACDE(s) are as follows:

a) Initialization: Same as that of classical DE.

*b) Clustering***:** After the completion of step (a); cluster centers are generated and membership matrix U_0 is randomly initialized for each population member. Individuals are then segregated into different clusters as per their membership value, i.e. individual *'i'* with maximum membership value for cluster *'k'* (μ_{ik} {∀k∈[1,2,....c] and ∀i ∈[1,2,.....,n]}) is assigned to that cluster.

The center of 'c' (number of clusters) clusters and the membership value of each individual (X_i^G) is updated (using eq-2 and eq-1 respectively) for each generation *G* until a stopping criteria as defined in step (7) is achieved. Thus for each generation, individuals are reassigned to different clusters depending upon updated membership matrix U^G (membership matrix U for generation G).

c) Mutation:

Mutation strategy for FCADE: In FCADE, the mutation strategy for each individual is decided according to the cluster to which the individual belongs. Two possibilities considered here are expressed through the equation below.

$$V = \begin{cases} \left(X_{ipos} + F * \left(Bestsol_{pos} - X_{ipos}\right) + F * \left(X_{nhd1_{pos}} - X_{nhd2_{pos}}\right)\right) \\ \qquad if\ X\ is\ in\ best\ individual's\ cluster \\ X_{rand1\_pos} + F * \left(X_{rand2_{pos}} - X_{rand3_{pos}}\right) \quad otherwise \end{cases}$$

Mutation strategies for FCADE1: For FCADE1 mutation strategy is designed by considering **three possibilities** (1) the current individual's cluster is the same as that of the best individual or (2) the difference between the membership value of the individual for both the clusters (boundary point) is less than the threshold value (10^-15) or **(3) if neither of the two possibilities are met.**

For FCADE 1, the strategy is expressed as:

$$V = \begin{cases} X_{ipos} + F * \left(Bestsol_{pos} - X_{ipos}\right) + F * \left(X_{nhd1_{pos}} - X_{nhd2_{pos}}\right) \\ \qquad if\ X\ is\ in\ best\ individual's\ cluster \\ X_{ipos} + F * \left(Min_{pos} - X_{ipos}\right) + F * \left(X_{nhd1_{pos}} - X_{nhd2_{pos}}\right) \\ \qquad if\ boundary\ point \\ X_{rand1\_pos} + F * \left(X_{rand2_{pos}} - X_{rand3_{pos}}\right) \quad otherwise \end{cases}$$

Mutation strategy for FCADE2: In FCADE2, mutation strategies are again defined by considering **three** possibilities (1) current individual's cluster is same as the best solution individual's cluster or (2) difference between the optimization cost of current individual and the best solution is less than the threshold value (10^-15) (3) neither of the two conditions are met. This strategy is defined as:

$$V = \begin{cases} X_{ipos} + F * \left(Bestsol_{pos} - X_{ipos}\right) + F * \left(X_{nhd1_{pos}} - X_{nhd2_{pos}}\right) \\ \qquad if\ X\ is\ in\ best\ individual's\ cluster \\ Bestsol_{pos} + F * \left(X_{nhd1_{pos}} - X_{nhd2_{pos}}\right) \\ \qquad if\ threshold\ condition\ holds \\ X_{rand1\_pos} + F * \left(X_{rand2_{pos}} - X_{rand3_{pos}}\right) \quad otherwise \end{cases}$$

For the above three strategies, the following notations are used :V: Mutant Vector; F: Scaling constant, randomly generated between 0.2 to 0.8; X_{ipos}: Position vector of the current individual.,$Bestsol_{pos}$: Position vector of the member with minimum objective function value present in the entire population known as $Bestsol$; Min_{pos} : Normalized value of the position of best solutions in both the clusters; X_{nhd1_pos}: Position vector of the random member present in the same cluster as a current individual known as neighbor1; X_{nhd2_pos}: Position vector of the random member present in the same cluster as the current individual known as neighbor2; X_{rand1_pos}: Position vector of a random member present in the population known as rand1; X_{rand2_pos}: Position vector of the random member present in the population is known as rand2,;: Position vector of the random member present in the population is known as rand3.

d) Cross-over: Crossover strategy is same for all three variants. The trial vector $T_j^G = \{t_{1,j}^G, t_{2,j}^G, \ldots, t_{D,j}^G\}$ is generated as $t_{i,j}^G = \begin{cases} v_{i,j}^G & if\ rand_j \le Cr \\ x_{i,j}^G & otherwise \end{cases}$

Cross-over rate(Cr) is updated as follows for each generationG:

$$Cr = \begin{cases} CRu, & where\ CRu = CRu + rand * 0.1 \quad If\ f(T) < f(X) \\ CRl, & where\ CRl = CRl + rand * 0.1 \quad Otherwise \end{cases}$$

where:

$f(T)$: Objective function value for trial vector T.

$f(X)$: Objective function value for current individual X.

CRl: Lower limit for Crossover rate set as 0.1.

CRu: Upper limit for Crossover rate - *CRu = 1- CRl*

e) Selection: same as classical DE for all three variants

IV. Experimental Settings and Numerical Results

CEC 2005[39] benchmark functions have been used to evaluate, validate and compare the performance of proposed variants with each-other and also with nine selected algorithms taken from literature. Overall 25 functions in 10 dimension (10D) and 30 dimension (30D) scenarios are considered. These include 5 unimodal functions, 7 basic multi-modal functions, 2 expanded multi-modal functions, 11 hybrid composite functions. Fig. 1 illustrates how FCADE attanined the optimum values in some functions

The maximum number of Function Evaluation (FE) is set to D*10,000 i.e. 100000 for 10D and 300000 for 30D. All algorithms were executed 25 times independently. The population size for each algorithm is set to 100 individuals for both 10D and 30D. FCADE algorithm and its variants FCADE1 and FCADE2 carry out fuzzy c-means clustering for grouping/ regrouping of the population with a cluster size of 2 and exponent of membership Z of 2. The proposed variants adopt an adaptive cross over-rate with the initial value of the crossover rate as 0.1 that is updated for each generation. The scaling factor for mutation is randomly generated between the range [0.2, 0.8].

For the purpose of comparison nine algorithms selected from literature include two basic variants of DE: DE/rand/1/bin[1], and DE/current-to-best/1/bin[1]; four most frequently referred adaptive variants of DE: jDE[28], SaDE[4], EPSDE[40], JADE[30] and three contemporary variants of Particle Swarm Optimization (PSO): APSO[31], CLPSO[31], OLPSO[31].

Table I shows mean error obtained through FCADE, FACDE1, FCADE2 algorithms. From Table □, it can be observed that FCADE outperforms the other two algorithms for 12 functions in 10D and 21 functions in 30D. FCADE performed better in comparison to FCADE1 and FCADE2 in terms of convergence also, which is visible from the graphs depicted through Figure 2. For all the functions considered here, the trend is similar however, only selected graphs are presented because of page limitation.

Tables II and III provides a comparison of FCADE with other nine algorithms for 10D and 30D problems respectively. From

the results, it can be observed that, for 10D problems, the FCADE algorithm outperforms jDE in 16 functions, SaDE in 12 functions, EPSDE in 8 functions and is better than JADE in 10 functions. We also observe that FCADE outperforms all the algorithms for the functions f18-f25, indicating the competence of FCADE for hybrid composition functions.

From Table 3, it can be observed that, for 30D problems, FCADE algorithm outperforms jDE in 13 functions, SaDE in 10 functions, EPSDE in 14 functions, JADE in 8 functions, APSO in 20 functions, CLPSO in 18 functions, OLPSO in 22 functions, DE/rand/1/bin in 22 functions and beat DE/current-to-best/1/bin in 21 functions. It is also observed that FCADE outperforms all the algorithms for the functions f17, f21-f24, indicating the competency of FACDE for hybrid composition functions in 30D as well.

Statistical Analysis: Friedman's ranking test is applied to analyze the performance of algorithms statistically for mean error. Table IV shows the ranking of algorithms for 30D. JADE got the best rank as it outperformed almost all the algorithms including FACDE, which secured the second rank.

TABLE I. COMPARISON OF RESULTS OBTAINED FCADE, FCADE1, FCADE2

Functions	10D			30D		
	FCADE	**FCADE 1**	**FCADE 2**	**FCADE**	**FCADE 1**	**FCADE 2**
1	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
2	0.00E+00	0.00E+00	0.00E+00	5.19E-07	6.60E-04	6.66E-04
3	8.61E+03	3.12E+04	2.12E+04	1.27E+05	1.59E+05	1.72E+05
4	0.00E+00	0.00E+00	0.00E+00	8.56E-04	7.37E-04	7.12E+00
5	0.00E+00	0.00E+00	0.00E+00	1.12E-01	1.35E+02	4.79E+01
6	8.35E-06	8.10E-02	2.04E-01	1.30E+01	3.94E+08	1.45E+08
7	1.27E+03	1.27E+03	1.27E+03	4.70E+03	4.70E+03	4.70E+03
8	2.03E+01	2.03E+01	2.04E+01	2.09E+01	2.09E+01	2.09E+01
9	8.67E-01	1.19E+01	1.37E+01	0.00E+00	3.99E+01	4.00E+01
10	1.32E+01	1.92E+01	1.96E+01	3.26E+01	1.12E+02	1.37E+02
11	1.57E+00	1.21E+00	1.58E+00	1.13E+01	2.41E+01	2.06E+01
12	6.15E+00	3.23E+02	2.21E+02	3.33E+03	4.69E+04	7.98E+04
13	1.91E+00	1.59E+00	1.55E+00	3.33E+00	1.32E+01	1.34E+01
14	3.25E+00	2.74E+00	2.76E+00	1.22E+01	1.25E+01	1.24E+01
15	2.51E+02	1.82E+02	2.02E+02	3.40E+02	4.37E+02	3.63E+02
16	1.09E+02	1.19E+02	1.18E+02	7.77E+01	2.51E+02	2.73E+02
17	1.51E+02	1.43E+02	1.38E+02	1.10E+02	3.06E+02	2.86E+02
18	4.14E+02	6.36E+02	6.48E+02	9.04E+02	9.43E+02	9.28E+02
19	4.52E+02	6.96E+02	6.34E+02	9.04E+02	9.23E+02	9.16E+02
20	4.64E+02	6.74E+02	6.03E+02	9.04E+02	9.34E+02	9.25E+02
21	5.00E+02	5.46E+02	6.35E+02	5.00E+02	5.55E+02	6.13E+02
22	7.76E+02	7.74E+02	7.76E+02	8.63E+02	9.06E+02	9.10E+02
23	5.78E+02	7.32E+02	7.16E+02	5.34E+02	7.54E+02	7.24E+02
24	2.00E+02	2.65E+02	2.00E+02	2.00E+02	4.42E+02	3.65E+02
25	2.00E+02	2.00E+02	2.43E+02	1.63E+03	1.63E+03	1.63E+03

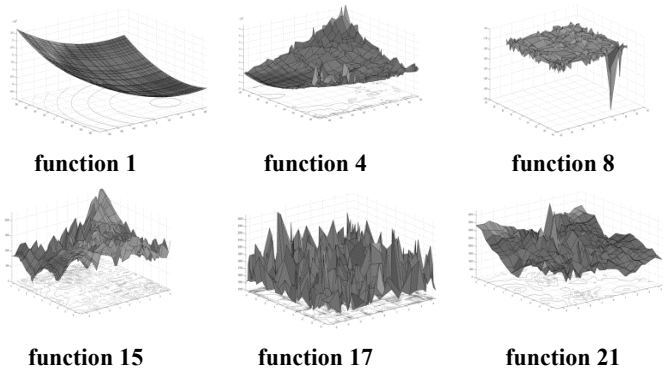
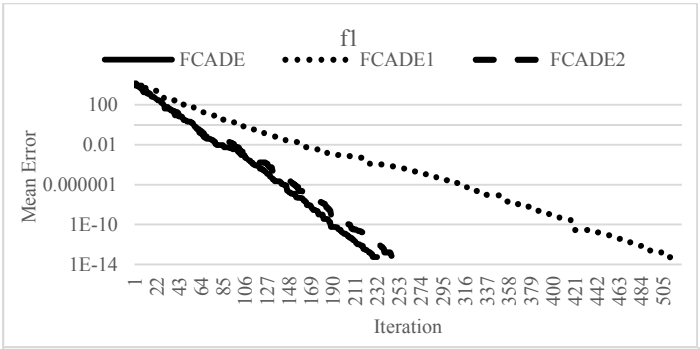


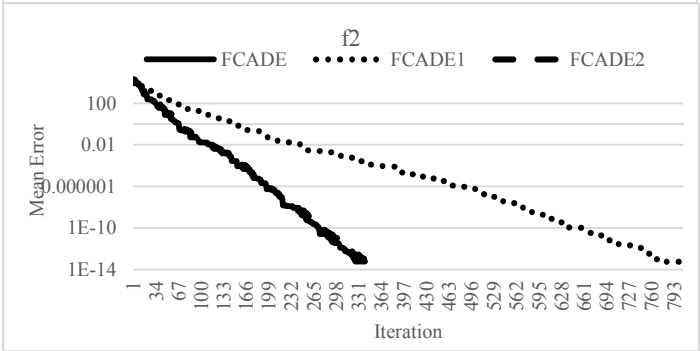
Figure 1. Optimum attained for selected functions through FCADE

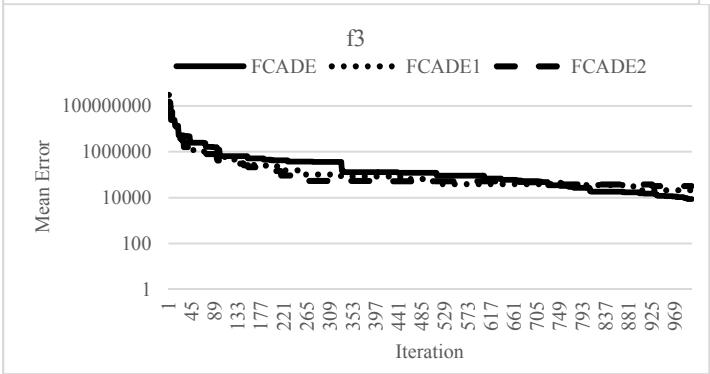
TABLE II. COMPARISON OF RESULTS USING DIFFERENT ALGORITHMS (10D)

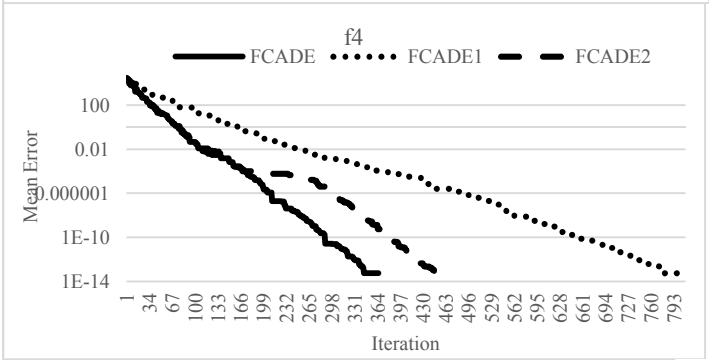
Algorithm / Function	f1		f2		f3	
	mean	stdev	mean	stdev	mean	stdev
jDE	0.00E+00	0.00E+00	6.01E-17	1.30E-16	4.15E-05	2.05E-04
SaDE	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.48E-25	6.03E-26
EPSDE	0.00E+00	0.00E+00	0.00E+00	0.00E+00	6.96E-25	1.73E-26
JADE	0.00E+00	0.00E+00	0.00E+00	0.00E+00	3.93E-26	2.65E-26
FCADE	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	8.61E+03	5.78E+03
Algorithm / Function	f4		f5		f6	
	mean	stdev	mean	stdev	mean	stdev
jDE	2.36E-16	2.86E-16	2.40E-12	3.43E-12	8.37E-02	2.33E-01
SaDE	0.00E+00	0.00E+00	1.10E-07	7.13E-08	1.30E+10	0.00E+00
EPSDE	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
JADE	0.00E+00	0.00E+00	0.00E+00	0.00E+00	6.39E-01	1.50E+00
FCADE	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**8.35E-06**	**2.95E-05**
Algorithm / Function	f7		f8		f9	
	mean	stdev	mean	stdev	mean	stdev
jDE	1.27E+03	9.28E-13	2.04E+01	6.07E-02	0.00E+00	0.00E+00
SaDE	1.71E-02	2.06E-02	2.04E+01	6.71E-02	0.00E+00	0.00E+00
EPSDE	3.94E-02	3.48E-02	2.04E+01	5.00E-03	0.00E+00	0.00E+00
JADE	1.27E+03	9.28E-13	2.03E+01	9.70E-02	0.00E+00	0.00E+00
FCADE	1.27E+03	9.28E-13	**2.03E+01**	**8.62E-02**	8.6E0-01	8.80E-01
Algorithm / Function	f10		f11		f12	
	mean	stdev	mean	stdev	mean	stdev
jDE	1.17E+01	2.46E+00	5.39E+00	1.71E+00	5.82E+01	2.68E+02

SaDE	6.71E+00	1.66E+00	4.50E+00	1.74E+00	2.51E+03	9.14E+02
EPSDE	3.30E+00	9.59E-01	4.16E+00	3.21E+00	5.00E+00	7.07E+00
JADE	3.81E+00	9.13E-01	4.16E+00	7.39E-01	6.23E+01	2.68E+02
FCADE	1.32E+01	9.13E+00	**1.57E+00**	**1.47E+00**	**6.14E+00**	**1.09E+01**
Algorithm / Function	f13		f14		f15	
	mean	stdev	mean	stdev	mean	stdev
jDE	4.25E-01	4.43E-02	3.31E+00	1.70E-01	2.27E+01	8.08E+01
SaDE	6.12E-02	9.09E-02	3.08E+00	2.65E-01	5.05E+01	1.32E+02
EPSDE	4.13E-01	1.08E-01	3.08E+00	8.31E-02	8.53E+01	1.49E+02
JADE	3.52E-01	6.53E-02	2.76E+00	2.40E-01	4.05E+01	1.10E+02
FCADE	1.91E+00	3.79E-01	3.25E+00	3.28E-01	2.51E+02	1.94E+02
Algorithm / Function	f16		f17		f18	
	mean	stdev	mean	stdev	mean	stdev
jDE	1.10E+02	6.73E+00	1.30E+02	1.03E+01	5.40E+02	2.55E+02
SaDE	9.65E+01	7.57E+00	1.13E+02	8.66E+00	7.60E+02	1.38E+02
EPSDE	9.76E+01	4.40E+00	1.21E+02	6.28E+00	6.37E+02	3.03E+02
JADE	9.67E+01	5.72E+00	1.03E+02	6.26E+00	6.99E+02	2.31E+02
FCADE	1.09E+02	2.02E+01	1.50E+02	1.93E+01	**4.14E+02**	**2.16E+02**
Algorithm / Function	f19		f20		f21	
	mean	stdev	mean	stdev	mean	stdev
jDE	5.80E+02	2.53E+02	6.80E+02	2.18E+02	5.28E+02	1.10E+02
SaDE	7.80E+02	1.00E+02	7.60E+02	1.38E+02	5.04E+02	1.74E+02
EPSDE	5.82E+02	3.11E+02	5.62E+02	2.96E+02	4.80E+02	6.10E+00
JADE	6.82E+02	2.19E+02	6.60E+02	2.29E+02	5.16E+02	2.15E+02
FCADE	**4.52E+02**	**2.45E+02**	**4.64E+02**	**2.47E+02**	**5.00E+04**	**1.74E-13**
Algorithm / Function	f22		f23		f24	
	mean	stdev	mean	stdev	mean	stdev
jDE	7.63E+02	3.62E+00	6.14E+02	1.38E+02	2.00E+02	0.00E+00
SaDE	7.23E+02	1.27E+02	7.26E+02	1.70E+02	2.00E+02	0.00E+00
EPSDE	7.70E+02	1.90E+01	2.13E+02	6.01E+01	2.00E+02	0.00E+00
JADE	7.49E+02	4.72E+00	7.39E+02	1.63E+02	2.00E+02	0.00E+00
FCADE	**7.75E+02**	**2.63E+01**	**5.78E+02**	**8.90E+01**	**2.00E+02**	**0.00E+00**
Algorithm / Function	f25		Overall Comparison (f1-f25)			
	mean	stdev	win"+"	loss"-"	neutral "="	
jDE	1.74E+03	4.63E+00	"16"	"6"	"3"	
SaDE	3.75E+02	2.49E+00	"12"	"9"	"4"	
EPSDE	4.41E+02	2.05E+00	"8"	"12"	"5"	
JADE	1.73E+03	5.73E+00	"10"	"8"	"7"	
FCADE	**2.00E+02**	**0.00E+00**				









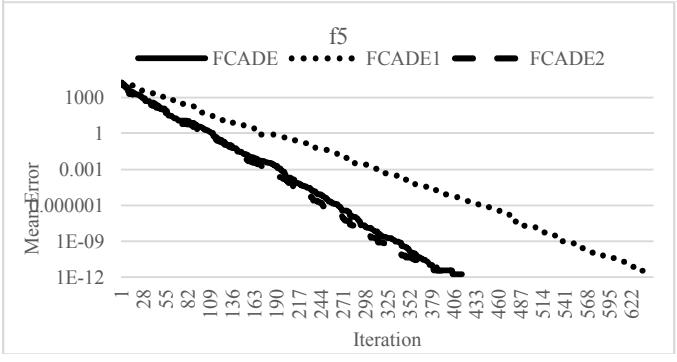


Figure 2. Convergence graph (10D) of benchmark functions (F1-F5) using FCADE, FCADE1 and FCADE2

TABLE III. COMPARISON OF RESULTS USING DIFFERENT ALGORITHMS (30D)

Algorithm/	f1		f2		f3	
Function	mean	stdev	mean	stdev	mean	stdev
jDE	0.00E+00	0.00E+00	2.46E-06	5.67E-06	1.52E+05	8.32E+04
SaDE	0.00E+00	0.00E+00	8.26E-18	2.48E-17	4.89E+04	2.53E+04
EPSDE	0.00E+00	0.00E+00	4.23E-26	4.07E-26	8.74E+05	3.28E+06
JADE	0.00E+00	0.00E+00	1.70E-28	1.74E-28	1.23E+04	9.75E+03
APSO	7.01E-14	2.45E-14	9.97E-13	1.79E-12	3.96E+05	1.59E+05
CLPSO	5.68E-14	0.00E+00	8.79E+02	1.79E+02	1.67E+07	4.66E+06
OLPSO	0.00E+00	0.00E+00	1.50E+01	1.23E+01	1.46E+07	5.33E+06
DE/rnd/1/bin	2.45E-05	3.48E-05	5.50E-05	1.28E-02	2.89E+05	1.94E+05
DE/current-to-best/1/bin	2.45E-05	3.48E-05	6.30E-08	1.29E-07	1.89E+05	1.04E+05
FCADE	**0.00E+00**	**0.00E+00**	5.19E-07	4.06E-07	1.26E+05	5.04E+04
Algorithm/	f4		f5		f6	
Function	mean	stdev	mean	stdev	mean	stdev
jDE	2.11E-02	2.66E-02	3.56E+02	4.45E+02	2.43E+01	2.67E+01
SaDE	1.69E-02	2.88E-02	8.85E+02	5.63E+02	3.96E+10	3.80E-05
EPSDE	3.49E+02	2.23E+03	1.40E+03	7.12E+02	6.38E-01	1.49E+00
JADE	1.23E-10	2.51E-10	3.00E-02	8.68E-02	7.65E+00	2.52E+01
APSO	7.23E+01	6.02E+01	5.85E+03	1.45E+03	6.94E+00	1.68E+01
CLPSO	6.61E+03	1.14E+03	3.86E+03	5.32E+02	5.10E+00	5.43E+00
OLPSO	2.26E+03	9.70E+02	3.28E+03	5.54E+02	2.63E+01	2.50E+01
DE/rnd/1/bin	5.04E-01	8.59E-01	1.27E+03	2.84E+02	2.78E+01	1.02E+01
DE/current-to-best/1/bin	1.07E-02	2.78E-02	5.76E+02	1.25E+02	9.27E+00	8.28E+00
FCADE	**8.56E-04**	**5.16E-04**	**1.12E-01**	**1.29E-01**	1.29E+01	3.32E+00
Algorithm/	f7		f8		f9	
Function	mean	stdev	mean	stdev	mean	stdev
jDE	4.70E+03	1.86E-12	2.09E+01	4.70E-02	0.00E+00	0.00E+00
SaDE	1.47E-02	1.27E-02	2.09E+01	3.81E-02	0.00E+00	0.00E+00
EPSDE	1.77E-02	1.34E-02	2.09E+01	5.81E-02	3.98E-02	1.99E-01
JADE	4.70E+03	1.86E-12	2.09E+01	1.95E-01	0.00E+00	0.00E+00
APSO	4.70E+03	2.34E-04	2.00E+01	2.97E-02	1.48E-13	5.90E-14
CLPSO	4.70E+03	6.78E-12	2.09E+01	5.46E-02	1.08E-11	1.02E-11
OLPSO	4.70E+03	1.61E-12	2.09E+01	6.90E-02	0.00E+00	0.00E+00
DE/rnd/1/bin	9.66E-01	9.66E-01	2.09E+01	6.25E-02	4.37E+01	9.73E+00
DE/current-to-best/1/bin	8.43E-01	9.14E-02	2.09E+01	5.01E-02	2.35E+01	8.37E+00
FCADE	4.70E+03	0.00E+00	2.09E+01	4.23E-02	0.00E+00	0.00E+00
Algorithm/	f10		f11		f12	
Function	mean	stdev	mean	stdev	mean	stdev
jDE	5.75E+01	9.22E+00	2.81E+01	1.56E+00	4.64E+03	5.43E+03
SaDE	3.98E+01	1.31E+01	1.86E+01	7.63E+00	1.39E+05	2.25E+04
EPSDE	5.36E+01	3.03E+01	3.56E+01	3.88E+00	3.58E+04	7.05E+03
JADE	2.73E+01	4.92E+00	2.54E+01	1.99E+00	6.33E+03	4.42E+03
APSO	1.50E+02	6.25E+01	2.78E+01	3.16E+00	1.27E+04	1.70E+04
CLPSO	1.14E+02	1.50E+01	2.70E+01	1.71E+00	2.81E+04	6.59E+03
OLPSO	1.10E+02	3.12E+01	2.55E+01	2.95E+00	1.33E+04	6.95E+03
DE/rnd/1/bin	1.56E+02	4.57E+01	3.26E+01	1.10E+01	8.43E+04	6.25E+04
DE/current-to-best/1/bin	1.96E+02	2.17E+01	3.17E+01	7.60E+00	9.84E+04	6.23E+03
FCADE	**3.26E+01**	**9.56E+00**	**1.13E+01**	**2.75E+00**	**3.33E+03**	**2.34E+03**
Algorithm/	f13		f14		f15	
Function	mean	stdev	mean	stdev	mean	stdev
jDE	1.70E+00	1.16E-01	1.29E+01	2.61E-01	4.04E+02	8.8.1E+01
SaDE	3.96E+00	3.89E-01	1.28E+01	2.74E-01	4.13E+02	3.34E+01
EPSDE	1.94E+00	1.46E-01	1.35E+01	2.09E-01	2.12E+02	1.98E+01
JADE	1.43E+00	1.20E-01	1.22E+01	3.22E-01	3.48E+02	9.18E+01
APSO	1.54E+00	4.05E-01	1.30E+01	5.24E-01	3.48E+02	1.50E+02
CLPSO	1.66E+00	5.68E-01	1.29E+01	1.72E-01	1.06E+02	5.34E+01
OLPSO	1.92E+00	3.28E-01	1.31E+01	2.57E-01	2.50E+02	9.21E+01
DE/rnd/1/bin	4.51E+00	2.27E+00	1.33E+01	3.48E-01	4.84E+02	2.15E+01
DE/current-to-best/1/bin	3.52E+00	2.27E+00	1.35E+01	3.48E-01	3.84E+02	5.14E+01
FCADE	3.33E+00	2.00E+00	**1.22E+01**	**3.47E-01**	3.40E+02	9.40E+01
Algorithm/	f16		f17		f18	
Function	mean	stdev	mean	stdev	mean	stdev
jDE	9.12E+01	6.75E+01	1.47E+02	6.45E+01	9.04E+02	7.28E-01
SaDE	5.33E+01	8.59E+00	6.46E+01	2.45E+01	8.43E+02	5.63E+01
EPSDE	1.22E+02	9.19E+01	1.69E+02	1.02E+02	8.20E+02	3.35E+00
JADE	6.30E+01	7.25E+01	1.45E+02	1.33E+02	9.05E+02	1.17E+00
APSO	3.22E+02	1.41E+02	3.12E+02	1.32E+02	9.45E+02	2.19E+01
CLPSO	1.88E+02	3.11E+01	2.40E+02	3.89E+01	9.12E+02	1.35E+00
OLPSO	1.32E+02	3.74E+01	1.89E+02	3.25E+01	9.10E+02	1.82E+00
DE/rnd/1/bin	2.82E+02	1.13E+01	3.09E+02	1.57E+01	9.13E+02	8.43E-01
DE/current-to-best/1/bin	2.23E+02	1.64E+02	2.35E+02	1.58E+02	9.50E+02	2.11E+01
FCADE	7.77E+01	3.62E+01	**1.10E+02**	**6.02E+01**	9.04E+02	7.04E-01
Algorithm/function	f19		f20		f21	
	mean	stdev	mean	stdev	mean	stdev

jDE	9.04E+02	1.30E+00	9.04E+02	8.17E-01	5.00E+02	1.16E-13
SaDE	8.58E+02	5.70E+01	8.59E+02	5.76E+01	5.00E+02	1.16E-13
EPSDE	8.21E+02	3.35E+00	8.22E+02	4.17E+00	8.33E+02	1.00E+02
JADE	9.04E+02	1.01E+00	9.05E+02	1.32E+00	5.12E+04	5.99E+01
APSO	9.45E+02	5.63E+01	9.36E+02	4.18E+01	7.66E+02	3.23E+02
CLPSO	9.13E+02	1.12E+00	9.13E+02	1.28E+00	5.00E+02	4.14E-13
OLPSO	9.07E+02	2.03E+01	9.07E+02	2.03E+01	5.00E+02	2.86E-13
DE/rnd/1/bin	9.20E+02	1.22E+00	9.13E+02	1.16E+00	5.81E+02	2.62E+01
DE/current-to-best/1/bin	9.52E+02	2.11E+01	9.41E+02	2.93E+01	8.32E+02	2.62E+02
FCADE	9.04E+02	1.09E+00	9.04E+02	9.37E-01	**5.00E+02**	**1.16E-13**
Algorithm/	f22		f23		f24	
Function	mean	stdev	mean	stdev	mean	stdev
jDE	8.75E+02	2.00E+01	5.34E+02	2.56E-04	2.00E+02	0.00E+00
SaDE	9.19E+02	1.19E+01	5.34E+02	1.42E-04	2.00E+02	0.00E+00
EPSDE	5.07E+02	7.26E+00	8.58E+02	6.82E+01	2.13E+02	1.52E+00
JADE	8.68E+02	1.62E+01	5.34E+02	2.19E-04	2.00E+02	0.00E+00
APSO	1.02E+03	5.65E+01	9.11E+02	3.04E+02	3.45E+02	3.76E+02
CLPSO	9.63E+02	1.36E+01	5.34E+02	1.39E-04	2.00E+02	6.19E-13
OLPSO	9.43E+02	1.35E+01	5.34E+02	3.59E-04	2.00E+02	2.89E-14
DE/rnd/1/bin	9.64E+02	1.14E+01	6.21E+02	3.06E+01	3.14E+02	3.22E+01
DE/current-to-best/1/bin	9.34E+02	4.15E+01	8.60E+02	2.88E+02	3.05E+02	3.56E+01
FCADE	**8.63E+02**	**8.78E+00**	**5.34E+02**	**3.75E-04**	**2.00E+02**	**6.74E-13**
Algorithm	f25			Overall Comparison (f1-f25)		
function	mean	stdev		win"+"	loss"-"	neutral "="
jDE	1.63E+03	4.00E+00		"13"	"1"	"11"
SaDE	2.17E+02	1.19E+00		"10"	"9"	"6"
EPSDE	2.13E+02	2.55E+00		"14"	"9"	"2"
JADE	1.63E+03	4.50E+00		"8"	"8"	"9"
APSO	1.70E+03	1.74E+01		"20"	"4"	"1"
CLPSO	1.65E+03	3.77E+00		"18"	"3"	"4"
OLPSO	1.64E+03	5.59E+00		"16"	"2"	"7"
DE/rnd/1/bin	9.86E+02	2.18E+01		"22"	"2"	"1"
DE/current-to-best/1/bin	9.68E+02	1.31E+01		"21"	"3"	"1"
FCADE	1.63E+03	4.67E+00				

TABLE IV. FRIEDMEN RANKING OF LISTED ALGORITHMS (30D)

Algorithm	Mean Rank
FCADE	**3.88**
jDE	4.88
SaDE	4
EPSDE	4.92
JADE	3.34
APSO	7.22
CLPSO	6.12
OLPSO	5.76
DE/rand	7.76
DE/best	7.12

V. CONCLUSION

The present study shows the implementation of Fuzzy C-means clustering for segregating the initial population of DE into different clusters which are further acted upon by adaptive crossover and cluster-specific mutation strategies. Three variants named FCADE, FCADE1 and FCADE2 are proposed which differ from each other on the basis of mutation strategies. Some conclusions that can be drawn from the present study are:

- Validation of the proposed variants on CEC 2005 benchmark problems, indicated FCADE to be the best performing variant in terms of solution quality as well as convergence. It also indicates that perhaps the incusion of three possibilities during mutation is not very effective.
- Further, comparison of FCADE with other algortihms indicates that FCADE is more suitable for composite functions.
- Statistical analysis in terms of mean error indicates that although FCADE is suitable for composite functions, its performance can be further improved.

REFERENCES

[1] R. Storn and K. Price, "Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces," *J. Glob. Optim.*, vol. 11, no. 4, pp. 341–359, 1997.

[2] K. Fleetwood, "An Introduction to Differential Evolution," *New ideas Optim.*, pp. 79–108, 1999.

[3] N. K. Madavan, "Multiobjective optimization using a Pareto differential evolution approach," *Proc. Congr. Evol. Comput. (CEC 2002)*, vol. 2, pp. 1145–1150, 2002.

[4] A. K. Qin and P. N. Suganthan, "Self-adaptive Differential Evolution Algorithm for Numerical Optimization," in *2005 IEEE Congress on Evolutionary Computation*, 2005, vol. 2, pp. 1785–1791.

[5] J. Liu and J. Lampinen, "A fuzzy adaptive differential evolution algorithm," *Soft Comput.*, vol. 9, no. 6, pp. 448–462, 2005.

[6] S. Kukkonen and J. Lampinen, "Constrained real-parameter optimization with generalized differential evolution," *2006 IEEE Congr. Evol. Comput. CEC 2006*, pp. 207–214, 2006.

[7] J. Tvrd??k, "Competitive differential evolution," *Mendel*, vol. 2006-Janua, no. January, pp. 7–12, 2006.

[8] E. Mezura-Montes, J. Velázquez-Reyes, and C. A. Coello Coello, "A comparative study of differential evolution variants for global optimization," *GECCO 2006 - Genet. Evol. Comput. Conf.*, vol. 1, pp. 485–492, 2006.

[9] A. Koh, "Solving transportation bi-level programs with differential evolution," *2007 IEEE Congr. Evol. Comput. CEC 2007*, pp. 2243–

2250, 2007.

[10] S. Rahnamayan, H. R. Tizhoosh, and M. M. A. Salama, "Opposition-based differential evolution (ODE) with variable jumping rate," *Proc. 2007 IEEE Symp. Found. Comput. Intell. FOCI 2007*, no. Foci, pp. 81–88, 2007.

[11] Z. Zhao, J. Yang, Z. Hu, and H. Che, "A differential evolution algorithm with self-adaptive strategy and control parameters based on symmetric Latin hypercube design for unconstrained optimization problems," *Eur. J. Oper. Res.*, vol. 250, no. 1, pp. 30–45, 2016.

[12] Pooja, P. Chaturvedi, and P. Kumar, "A Cultivated Differential Evolution Variant for Molecular Potential Energy Problem," *Procedia Comput. Sci.*, vol. 57, pp. 1265–1272, 2015.

[13] Y. Chen, W. Xie, and X. Zou, "A binary differential evolution algorithm learning from explored solutions," *Neurocomputing*, vol. 149, no. PB, pp. 1038–1047, 2015.

[14] K. P. Chandar and T. S. Savithri, "3D Face Model Estimation based on Similarity Transform using Differential Evolution Optimization," *Procedia Comput. Sci.*, vol. 54, pp. 621–630, 2015.

[15] Y. Zhou, X. Li, and L. Gao, "A differential evolution algorithm with intersect mutation operator," *Appl. Soft Comput. J.*, vol. 13, no. 1, pp. 390–401, 2013.

[16] D. Datta and S. Dutta, "A binary-real-coded differential evolution for unit commitment problem," *Int. J. Electr. Power Energy Syst.*, vol. 42, no. 1, pp. 517–524, 2012.

[17] G. A. S. Segundo, R. A. Krohling, and R. C. Cosme, "A differential evolution approach for solving constrained min-max optimization problems," *Expert Syst. Appl.*, vol. 39, no. 18, pp. 13440–13450, 2012.

[18] C. W. Chiang, W. P. Lee, and J. S. Heh, "A 2-Opt based differential evolution for global optimization," *Appl. Soft Comput. J.*, vol. 10, no. 4, pp. 1200–1207, 2010.

[19] J. Tvrdík, "Adaptation in differential evolution: A numerical comparison," *Appl. Soft Comput. J.*, vol. 9, no. 3, pp. 1149–1155, 2009.

[20] Ç. Balkaya, Y. L. Ekinci, G. Göktürkler, and S. Turan, "3D non-linear inversion of magnetic anomalies caused by prismatic bodies using differential evolution algorithm," *J. Appl. Geophys.*, vol. 136, pp. 372–386, 2017.

[21] S. Rahnamayan, H. R. Tizhoosh, and M. M. Salama, "Opposition-based differential evolution," *Stud. Comput. Intell.*, vol. 143, no. 1, pp. 155–171, 2008.

[22] J. Teo, "with Self-adaptive Populations," pp. 1284–1290, 2005.

[23] J. Teo, "Exploring dynamic self-adaptive populations in differential evolution," *Soft Comput.*, vol. 10, no. 8, pp. 673–686, 2006.

[24] W. Deng, X. Yang, L. Zou, M. Wang, Y. Liu, and Y. Li, "An improved self-adaptive differential evolution algorithm and its application," *Chemom. Intell. Lab. Syst.*, vol. 128, pp. 66–76, 2013.

[25] Y. Sun, Y. Li, G. Liu, and J. Liu, "A novel differential evolution algorithm with adaptive of population topology," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2012, vol. 7473 LNCS, pp. 531–538.

[26] J. Aalto and J. Lampinen, "A population adaptation mechanism for differential evolution algorithm," in *Proceedings - 2015 IEEE Symposium Series on Computational Intelligence, SSCI 2015*, 2015, pp. 1514–1521.

[27] S. M. Islam, S. Das, S. Ghosh, S. Roy, and P. N. Suganthan, "An adaptive differential evolution algorithm with novel mutation and crossover strategies for global numerical optimization," *IEEE Trans. Syst. Man, Cybern. Part B Cybern.*, vol. 42, no. 2, pp. 482–500, 2012.

[28] M. Mernik, J. Brest, and V. Zumer, "Self-Adapting Control Parameters in Differential Evolution : A Comparative Study on," vol. 10, no. 6, pp. 646–657, 2006.

[29] R. Mallipeddi and P. N. Suganthan, "Differential evolution algorithm with ensemble of parameters and mutation and crossover strategies," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 6466 LNCS, pp. 71–78, 2010.

[30] J. Zhang and A. C. Sanderson, "JADE: Adaptive differential evolution with optional external archive," *IEEE Trans. Evol. Comput.*, vol. 13, no. 5, pp. 945–958, 2009.

[31] Y. F. Li, Z. H. Zhan, Y. Lin, and J. Zhang, "Comparisons study of APSO OLPSO and CLPSO on CEC2005 and CEC2014 test suits," *2015 IEEE Congr. Evol. Comput. CEC 2015 - Proc.*, pp. 3179–3185, 2015.

[32] P. Bujok and J. Tvrdík, "New Variants of Adaptive Differential Evolution Algorithm with Competing Strategies," *Acta Electrotech. Inform.*, vol. 15, no. 2, pp. 49–56, 2015.

[33] A. W. Mohamed, "An improved differential evolution algorithm with triangular mutation for global numerical optimization," *Comput. Ind. Eng.*, vol. 85, pp. 359–375, 2015.

[34] W. Gong, Z. Cai, and Y. Wang, "Repairing the crossover rate in adaptive differential evolution," *Appl. Soft Comput. J.*, vol. 15, pp. 149–168, 2014.

[35] R. Gämperle, S. D. Müller, and P. Koumoutsakos, "A parameter study for differential evolution," *Adv. Intell. Syst. fuzzy Syst. Evol. Comput.*, vol. 10, pp. 293–298, 2002.

[36] J. Liu and J. Lampinen, "On setting the control parameter of the differential evolution method," in *Proceedings of the 8th International Conference on Soft Computing (MENDEL)*, 2002, pp. 11–18.

[37] J. C. Dunn, "A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters," *J. Cybern.*, vol. 3, no. 3, pp. 32–57, Jan. 1973.

[38] J. C. Bezdek, "A convergence theorem for the fuzzy subspace clustering (FSC) algorithm," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 6, pp. 1939–1947, 1980.

[39] P. N. Suganthan *et al.*, "Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization," *Tech. Report, Nanyang Technol. Univ. Singapore, May 2005 KanGAL Rep. 2005005, IIT Kanpur, India*, no. May 2014, 2005.

[40] R. Mallipeddi and P. N. Suganthan, "Differential evolution algorithm with ensemble of parameters and mutation and crossover strategies," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 6466 LNCS, no. November 2014, pp. 71–78, 2010.

[41] J. Liu & J. Lampinen, (2005). "A fuzzy adaptive differential evolution algorithm". *Soft Computing*, *9*(6), 448-462.