

# Differential Evolution with Generalized Mutation Operator

H. Sharifi Noghabi, *student member, IEEE*, H. Rajabi Mashhadi, and K. Shojaei, *Member, IEEE*

**Abstract**—Differential Evolution (DE) proved to be one of the most successful evolutionary algorithms for global optimization purposes in continuous problems. The core operator in DE is mutation which can provide the algorithm with both exploration and exploitation. In this article, a new notation for DE is proposed which has a formula that can be utilized for generating and extracting novel mutations and by applying this new notation, four novel mutations are proposed. More importantly, by combining these novel trial vector generation strategies and four other well-known ones, we proposed Generalized Mutation Differential Evolution (GMDE) that takes advantage of two mutation pools that have both explorative and exploitative strategies inside them. Results and experimental analysis are performed on CEC2005 benchmarks and the results stated that GMDE is surprisingly competitive and significantly improved the performance of this algorithm.

**Index Terms**—Generalized Mutation Differential Evolution, Differential Evolution, Mutation operator, Global Numerical Optimization, Trial Vector Generation Strategy.

## I. INTRODUCTION

EVOLUTIONARY algorithms such as Genetic Algorithm (GA) or Evolutionary Strategy (ES) are stochastic search methods that are generally inspired from natural phenomena like natural selection or survival of the fittest [1, 2]. In recent years, evolutionary algorithms were applied to solve numerous real world problems and among these algorithms Differential Evolution (DE) has received so much attention in both developing itself and application areas [3]. DE was originally proposed in 1995 by Storn and Price for solving Chebyshev polynomial fitting problem [4]. DE has numerous advantages in comparison with other methods such as simple structure and implementation, low computational complexity, high performance, being robust and the fact that it is intersection of classic methods such as Controlled Random Search and evolutionary methods such as GA and Simulated Annealing (SA) [5]. More information is available in two good reviews [3, 6].

Generally, an evolutionary algorithm such as DE starts its optimization process with an initial population, this population

is evaluated by cost/fitness function and better individuals will move to the next generation [7]. Through this process, individuals are evolved with mutation and crossover operators and this evolution will be continued until the algorithm reaches the acceptable answer or a stopping condition [3]. Mutation operator is utilized in order to generate a *donor* vector and after that the generated vector is used in crossover for generating a new trial vector or a candidate individual [8]. In fact, the goal of mutation operator is to provide the algorithm with more diversity, exploration and reliability, therefore, mutation is the core operator in DE [5]. In this operator, three vectors are chosen randomly, however, because of this randomness, one of the main drawbacks of DE is its weak exploitation capability [9]. The most important issue regarding evolutionary algorithms is how to establish a tradeoff between exploration and exploitation. Exploration or reliability helps the algorithm to explore all of the problem space to find promising regions and exploitations enhances the process of finding the optimum in that region [10]. Using inappropriate vector generation strategy or parameters setting can lead to over-explorative or over-exploitative behaviors just like random search or greedy search [11]. Although, an over-explorative mutation can determine most of the problem space but without exploitation pressure there would be no effective convergence. Therefore, providing the balance between exploration and exploitation is center of studies regarding most of the evolutionary algorithms such as DE [12].

This paper investigates whether the performance of DE can be improved by generalizing diverse mutations that have both explorative and exploitative properties. In this paper, we proposed a new notation for DE that can generates novel mutations for this algorithm and fills the empty places of the stated tradeoff. Synergism of these generated mutation and previous mutations in two mutation pools significantly improved the performance of DE in almost all of the cases.

The rest of the paper is organized as follow: Section II is about categorizing mutations in DE literature, Section III describes the proposed method and Section IV presents results and experimental analysis and finally Section V concludes the paper.

H. Sharifi Noghabi is with Department of Computer Engineering at Ferdowsi University of Mashhad, Iran. Email: hsharifi@ieee.org; hsharifi@stu.um.ac.ir

H. Rajabi Mashhadi is professor at Departments of Electrical and Computer Engineering at Ferdowsi University of Mashhad, Iran. Email: h\_mashhadi@um.ac.ir

Kambiz Shojaei is with Department of Computer Engineering at Ferdowsi University of Mashhad, Iran. Email: shojaei@stu.um.ac.ir

All authors are member of Center of Excellence on Soft Computing and Intelligent Information Processing (SCIIP) located at Faculty of Engineering at Ferdowsi University of Mashhad, Iran.

## II. RELATED WORK

Mutation is the core operator in DE and therefore, so many efforts are performed to improve and develop this operator [3]. In fact, mutation is the key to distinguish between variants of DE by applying DE/X/Y/Z notation where DE denotes differential evolution, X denotes the target or base vector, Y denotes number of randomly selected difference vectors and Z indicates type of crossover operator [3, 6]. Since in this paper we applied binomial crossover (bin), Z is omitted from this notation in further usage of it. In this section, we categorized previous efforts on mutation into four generations and discussed each of them briefly.

### A. First generation: Storn and Price mutations

In honor of the researchers that proposed DE for the first time, we entitled this generation Storn and Price. In this generation, mutations exist that have been widely used in DE literature so far. Most of these mutations are proposed by Storn and Price and they are following the classic notation of DE/X/Y. Some of the well-known mutations of this family are listed as follow [3, 5]:

DE/rand/1

$$X_{new,g} = X_{r1,g} + F \cdot (X_{r2,g} - X_{r3,g}) \quad (1)$$

DE/best/1

$$X_{new,g} = X_{best} + F \cdot (X_{r1,g} - X_{r2,g}) \quad (2)$$

DE/rand/2

$$X_{new,g} = X_{r1,g} + F \cdot (X_{r2,g} - X_{r3,g}) + F \cdot (X_{r4,g} - X_{r5,g}) \quad (3)$$

DE/best/2

$$X_{new,g} = X_{best} + F \cdot (X_{r1,g} - X_{r2,g}) + F \cdot (X_{r3,g} - X_{r4,g}) \quad (4)$$

DE/current-to-best/1

$$X_{new,g} = X_{current,g} + F \cdot (X_{best} - X_{current,g}) + F \cdot (X_{r2,g} - X_{r3,g}) \quad (5)$$

DE/rand-to-best/1

$$X_{new,g} = X_{r1,g} + F \cdot (X_{best} - X_{r1,g}) + F \cdot (X_{r2,g} - X_{r3,g}) \quad (6)$$

DE/current-to-rand/1 ( $F' = ki \cdot F$ )

$$X_{new,g} = X_{current,g} + k \cdot (X_{r1,g} - X_{current,g}) + F' \cdot (X_{r2,g} - X_{r3,g}) \quad (7)$$

Where,  $X_{new,g}$  is the generated vector by mutation operator,  $X_{best}$  indicates the best individual,  $X_{current,g}$  indicates the current member of the population at generation  $g$ ,  $r_1 \neq r_2 \neq r_3 \neq r_4 \neq r_5 \neq current \neq new$  are randomly selected indices and  $F$  is the scaling factor. Please note that in equations (1)-4) and (6) the *current* index is not present,

however the  $r_1 \dots r_5$  need to be different from *new*.

Some of the mutations in this generation are explorative such as DE/rand/1 or DE/rand/2 and some others are more exploitative such as DE/best/1 or DE/current-to-best/1.

The main drawback of DE is its exploitation which means there is no adequate tradeoff in exploration and exploitation in DE because it explores the problem space more and the exploitation phase in it is not well enough [13]. The importance of this generation is the fact that, these mutations are foundation of DE and even today they are acting as criteria for evaluation of new variants of DE. The next generation is dealing with efforts to provide the algorithm with a balance between exploration and exploitation.

### B. Second generation: A quest for better tradeoff

In this family, the simple structure of the mutation operator according to its classic notation is deformed. This deformation occurred in hope of establishing a better tradeoff between exploration and exploitation. Mutation such as DE/rand/2/dir, trigonometric mutation, DE/rand/1/Either-or-algorithm and DEGL are some of the examples of this generation [12, 14, 15]. The other aspect of this generation is applying concepts such as niching and crowding in DE especially when it comes to multimodal problems [16]. In addition to fitness sharing approaches, in this generation, dividing population to sub-populations and performing diverse mutations on them can be considered another part of this generation [17-20].

The next generation is about an important step in evolution of every evolutionary algorithm.

### C. Third generation: Adaptive mutations

One of the most important developments on every evolutionary computation algorithm is parameters control and adaptation [21]. Controlling parameters proved to be significantly influential on general performance of an evolutionary computation method. Similar to other algorithms, DE is receiving a lot of attention regarding its parameters and how to control them [3]. Since scaling factor has impact on performance of mutation operator in DE directly, this generation is dealing with efforts and tries to control this parameter. This family of DE starts from deterministic parameter control and fuzzy logic [22] to adaptive and self-adaptive parameter control methods such as jDE [23], JADE [24], CoDE [25], SaDE [26], etc. which are well-known members of this generation.

It is important to note that algorithms such as SaDE and CoDE also take advantage of mutation pool. SaDE selects one mutation from its pool according to success and failure rates of that mutation in previous usages but CoDE performs all of the mutations in its pool and keeps the best obtained result from them.

The next generation is dealing with making mutation operator in DE intelligent in its selection of parent vectors.

### D. Fourth generation: Intelligent Selection

Usually, Parent vectors in DE are selected randomly from population, however, since good species in the population have good and valuable information, they are more likely to guide

the other individuals in the population [13]. Based on this fact in nature, this generation tries to select parent vector intelligently by a meaningful criterion. Generally, this criterion is from fitness space such as Ranking-based mutation [13] or it is adopted from design space such Proximity-based mutation [27].

Another algorithms such as DERL [28], JADE [24], DE based on FER [29] can be considered other members of this generation.

The importance of this generation is the fact that this family provides DE with more exploitation which is helpful regarding the main drawback of it.

### III. PROPOSED METHOD

In this section, we proposed a novel notation for DE which is capable of generating and extracting new mutation operators similar to the first generation mutations. The classic notation does not have such a capability and it is limitative. New mutations can be obtained by following equation:

$$V_i = X_{t_1} + \sum_{j=1}^n F_j * (X_{t_{2j}} - X_{t_{2j+1}}) \quad (8)$$

Where,  $V_i$  is donor vector,  $n$  is number of difference vectors,  $F_j$  is the scaling factor of the  $j$ -th difference vector and  $X_{ti}$  is a member of population. Determination of  $X_{ti}$  is extremely related to the problem and purpose of the operator. There are numerous options for each of  $X_{ti}$ s in this equation including [3], best individual, current member, random member, top  $p\%$  member [24], worst  $p\%$  member, tournament winner (or any other selection method) [28], a member from a neighborhood [20] or an intelligently selected member according to fourth generation. Therefore, via playing with these building blocks and specifying each member a novel mutation can be obtained. For instance, if exploration is crucial for a problem, placing an explorative modules such as randomly selected members makes the generated mutation explorative or if exploitation is required, utilizing exploitative modules such as best or top  $p\%$  makes the new mutation exploitative.

In this paper, all of the mutations for  $n = 1$  and  $n = 2$  are generalized and evaluated as a preprocess phase and then the best ones are selected for second round evaluation. In the second round, selected mutations take advantage of higher function evaluation and a self-adaptive parameter control mechanism. The best mutations in the second round are designated for Generalized Differential Evolution (GMDE) mutation pools. These top mutations are categorized into two mutation pools and at each generation only mutations of one of these pools are executed and the best generated individual will be the candidate for entering to the next generation. In order to determine the mutation pool, a Set Selection Rate (SSR) parameter is defined. Designated mutations should be categorized into two pools in a way that both of the pools have explorative and exploitative mutations. This issue is important because it provides the algorithm with both exploration and exploitation for the entire process of optimization.

Therefore, the new notation for DE is presented as follow:

$$GDE(T, F, d_1, d_2) \quad (9)$$

In this notation, *GMDE* denotes, Generalized Differential Evolution,  $F$  denotes types of scaling factor for instance in case of applying jDE parameter control for all of the difference vectors it would be “jDE” instead of  $F$  which size of  $F$  is  $n*1$ . For difference vectors,  $d_1$  and  $d_2$  are proposed and both of them are  $n*1$ .  $d_1$  indicates the first element of all of the difference vectors and  $d_2$  indicates the second elements of these vectors.  $T$  in the proposed notation, denotes the target or base vector of the mutation, in fact,  $X_{t1}$  in Eq. (8) determines  $T$  in Eq. (9) and  $X_{t2j}$  and  $X_{t2j+1}$  are specifying difference vectors for Eq. (9). For instance, DE/rand/1 mutations in classic notation that lets use JADE as a parameter control is following in the new notation: *GMDE(rand, "JADE", rand, rand)*.

Algorithm 1 is pseudocode of the GMDE.

Algorithm 1 GMDE:

- 1: Initialization of parameters such as SSR.
- 2: Evaluation the initial population.
- 3: **while** (termination criterion is not satisfied)
- 4:   select the mutation pool according to SSR.
- 5:   perform mutations of the selected pool.
- 6:   perform crossover for all of the generated donor vectors.
- 7:   select the best trial vector generated by crossover operator.
- 8:   **if** (trial vector < current member of the population) **then**
- 9:     enter the trial vector to the population.
- 10:   **endif**
- 11: **end while**

### IV. RESULTS AND EXPERIMENTAL ANALYSIS

Comprehensive and extensive experiments are performed in order to study the performance of the proposed notation and GMDE. In this section, first parameters settings and benchmark functions are introduced, then, the preprocess phase is discussed in detail. After preprocessing, second round evaluation is presented and finally the performance of GMDE is investigated.

#### A. Benchmark functions

We utilized 25 benchmark functions from CEC2005 competition [30]. These functions are categorized into three groups: 1) F1-F5: these functions are unimodal, 2) F5-F12: these functions are basic multimodal functions, 3) F13 and F14 are expanded multimodal functions and finally 4) F15-F25 are hybrid composition functions.

#### B. Parameters settings

Parameters are extremely influential on the performance of the algorithm, in all cases, we applied jDE parameter control to have fair situation for all of the mutations and algorithms except when the algorithms has its own parameter control methods

such as SaDE or CoDE. jDE [23] controls scaling factor and crossover rate as follow:

$$F_{i,G+1} = \begin{cases} F_l + rand_1 * F_u & rand_2 < \tau_1 \\ F_{i,G} & \text{otherwise} \end{cases} \quad (10)$$

$$Cr_{i,G+1} = \begin{cases} rand_3 & rand_4 < \tau_1 \\ Cr_{i,G} & \text{otherwise} \end{cases} \quad (11)$$

Where,  $rand_{i, i \in 1 \dots 4} = \{rand \in [0,1]\}, \tau_1 = \tau_2 = 0.1,$   
 $F_l = 0.1, F_u = 0.9$

This approach makes  $F \in [0.1, 0.9]$  and  $Cr \in [0, 1]$ .

The other parameters are initialized as follow:

$NP$  as number of population=50; Maximum run=50;  $D$  as number of dimensions=30 and maximum number of function evaluation= $D * 10\ 000$ . These settings are applied for second round evaluation and GMDE phases and for preprocess phase another settings are used as follow:

$NP=50$ ; maximum number of function evaluation=10 000;  $D=10$ ; Maximum run=100;  $F=0.5$  and  $Cr$  as crossover rate=0.9;

### C. Preprocess phase

In this section, diverse mutations generated by the proposed method are swept with stated settings in order to find potential mutations that can be useful for GMDE. In this phase, we set  $n=1$  and  $n=2$  but for simplicity only randomly selected member, the best member and current member of population are considered for blocks of the mutations. In addition to this assumption, in case of  $n=2$ , the second difference vector is also considered to be just random. The results for preprocess phase are presented in TABLE I through TABLE VI. In these tables,  $X_{ii}, i = 1, 2, \dots, 5$  is determined according to Eq. (8) and effective mutations are specified in **boldface** and these mutations are considered for second round evaluation.

In case of  $n=1$ , mutations 1, 4, 6, 10, 12 and 16 have the best performance among other mutations. Mutation 1 was successful in 4 functions but was not very successful in others. Mutation 4 found the best answer in function 11 and achieved acceptable results in 10 other functions but was not entirely effective for functions such as 7, 8, 10 and 14. Mutation 6 obtained good results in functions 1 through 4 and also 6 and 16. Mutation 10 had the best result in functions 2, 5 and 6 and moreover, achieved good results in functions 1, 3, 4, 13 and 16. In comparison with other mutations, mutation 12 found the best outcome for functions 1, 4, 9 and 13 and also obtained good ones for functions 2, 3, 6 and 16. Finally, mutation 16 achieved the best result among other methods for 3 functions and also was successful for 6 other functions. These 6 mutations mostly were successful for unimodal functions and rarely effective for multimodal functions which was expectable due to lack of exploitation in DE. This drawback is significantly solved in mutations 4, 10 and 12 which take advantage of the best member in their structure. Interestingly, mutation 4 that has the best member in its difference vector appears to be strongly competitive in comparison with well-known mutation 10 (DE/best/1). Among these mutations, 1 and 10 were presented in DE literature before but 4, 6, 12 and 16 are suggested by this article.

In case of  $n=2$ , a second difference vector with randomly selected individuals is added to the mutation operators of  $n=1$ . This means that previous mutations are provided with more exploration and reliability. From 27 possible mutations in  $n=2$ , 11 mutations are selected for second round evaluation. Among them, 1, 4, 10, 21 and 24 were previously presented in other papers and mutations 11, 12, 13, 15, 16 and 17 are suggested in this paper. According to tables III and VI, using the current member of population is not achieving to competitive results. After determination of elite mutations, in the second round evaluation these mutations are provided with more function evaluation and parameter adaptation based on jDE as stated in the previous section. The results of second round evaluation are presented in TABLE VII and TABLE VIII for  $n=1$  and  $n=2$  respectively. In these tables results are evaluated by Wilcoxon Signed-rank test for  $\alpha = 0.05$  and according to this test  $w$  indicates that the proposed mutation by Eq. (8) is significantly better than the compared mutation,  $l$  indicates that the proposed mutation is worse and  $t$  indicates that it is equal to the compared mutation.

For the second round evaluation of  $n=1$ , according to TABLE VII, the proposed mutations won in 73 cases, lost in 63 cases and performed equally in 13 cases. Mutation 4 obtained the best result comparing to mutations 1 and 10. This mutation won in 25, lost in 15 and tied in 5 comparing to previously suggested mutations (DE/rand/1 and DE/best/1). Mutation 12 in the same comparison won in 21, lost in 25 and tied in 4 cases and finally mutation 6 won in 22 lost in 24 and tied in 4 functions. Therefore, for mutations with three vectors (one vector in the base and two in difference part) placing the best member of population in the first part of difference vector is leading to better results. In addition to this, applying current member in these family of mutations appears to be ineffective. Mutations 10 and 12 were not successful comparing to the other three mutation, however, they achieved to the best results for functions 3, 14 and 25. Mutation 1 was superior for hybrid functions (F15-F25) but it lost its superiority to mutation 4 in multimodal functions (F6-F14). Generally, the three proposed mutation (4, 6 and 12) were significantly better comparing to mutation 10 (DE/best/1) in both hybrid and multimodal functions, however, comparing to mutation 1 (DE/rand/1) they were better mostly in unimodal (F1-F5) and multimodal functions.

For second round evaluation of  $n=2$ , Mutations with 5 vectors (one vector in the base, and four ones in two difference vectors) are studied in this part of evaluation. According to TABLE VIII, mutations 1 and 4 were significantly better than proposed mutations for  $n=2$ . According to this table, among the proposed mutations, 12, 16 and 15 had the most wins, however, this performance was not well enough. From achieving the best result point of view, mutations 15 and 17 were the worst without any achievement and mutations 11 and 13 also only achieved the best results for two functions (functions 11 and for mutation 13 and function 14 for mutation 11). In order to save space, the details of the results for  $n=2$  mutations are not stated. Both scenarios contain cases which are obviously ineffective such as mutation 5 in TABLE I or mutation 14 in TABLE III, however, we consider them in comparison in order to have a complete study.

#### D. Generalized Mutation Differential Evolution (GMDE)

So far 54 mutations were examined and evaluated in preprocess phase for  $n=1$  and  $n=2$  according to Eq. (8) and 16 of them were selected for the second round evaluation with stated parameters settings. According to the new notation in (9), these mutations are as follow (in “()” the old notation of those that presented before is stated):

- 1)\* GMDE#1( $rand, jDE, rand, rand$ ). ( $rand/1$ )
- 2) GMDE#2( $best, jDE, rand, rand$ ). ( $best/1$ )
- 3) GMDE#3( $rand, jDE, best, current$ ).
- 4)\* GMDE#4( $rand, jDE, best, rand$ ).
- 5) GMDE#5( $best, jDE, rand, current$ ).
- 6)\* GMDE#6( $rand, jDE, \begin{matrix} rand & rand \\ rand' & rand \end{matrix}$ ). ( $rand/2$ )
- 7)\* GMDE#7( $rand, jDE, \begin{matrix} best & rand \\ rand' & rand \end{matrix}$ ). ( $rand\text{-to}\text{-best}/1$ )
- 8) GMDE#8( $best, jDE, \begin{matrix} rand & rand \\ rand' & rand \end{matrix}$ ). ( $best/2$ )
- 9) GMDE#9( $best, jDE, \begin{matrix} rand & rand \\ best' & rand \end{matrix}$ ).
- 10)\* GMDE#10( $best, jDE, \begin{matrix} rand & rand \\ current' & rand \end{matrix}$ ).
- 11)\* GMDE#11( $best, jDE, \begin{matrix} best & rand \\ rand' & rand \end{matrix}$ ).
- 12) GMDE#12( $best, jDE, \begin{matrix} best & rand \\ current' & rand \end{matrix}$ ).
- 13)\* GMDE#13( $best, jDE, \begin{matrix} current & rand \\ rand' & rand \end{matrix}$ ).
- 14) GMDE#14( $best, jDE, \begin{matrix} current & rand \\ best' & rand \end{matrix}$ ).
- 15)\* GMDE#15( $current, jDE, \begin{matrix} rand & rand \\ current' & rand \end{matrix}$ ). ( $current\text{-to}\text{-rand}/1$ )
- 16) GMDE#16( $current, jDE, \begin{matrix} best & rand \\ current' & rand \end{matrix}$ ). ( $current\text{-to}\text{-best}/1$ )

Those 10 mutations that specified in boldface were able to find the best result for at least one function comparing to the other methods. Among them, 8 mutations that specified with “\*” are selected for mutation pools. In the case of GMDE#9 and GMDE#3, although they were able to achieve the best outcome in 1 and 3 functions respectively, for GMDE#9 the standard deviation between all the mutations was near zero which means all the methods were performed almost the same and for GMDE#3, other mutations were able to achieve the same result as it did, consequently, in order to reduce these overlaps, GMDE#3 was not considered for mutation pools. The top 8 mutations are grouped into two mutation pools as follow:

$$\begin{cases} \text{Pool 1: } GDE \#4, GDE \#6, GDE \#11 \text{ and } GDE \#15 \\ \text{Pool 2: } GDE \#1, GDE \#7, GDE \#10 \text{ and } GDE \#13 \end{cases} \quad (12)$$

Each pool contains both explorative and exploitative mutations and SSR determines the designated pool in each generation. In this article, being explorative or exploitative is simply defined

by the structure of mutations. If a mutation has best or  $p\%$  best components it is exploitative and if it has random modules it is explorative. SSR is initialized to 0.5, hence, both pools have 50% chance for selection as follow:

$$\begin{cases} \text{if } rand \leq SSR & \text{Select pool \#1} \\ \text{if } rand > SSR & \text{Select pool \#2} \end{cases} \quad (13)$$

Where,  $SSR = 0.5$  is set selection rate and  $rand \in (0,1)$  is a randomly selected number.

The main advantage of such a method is synergism between old and new mutations and the fact that by applying the proposed notation, empty places are filled and competitive mutations such as GMDE#4 are obtained. At each generation, all of the mutations of the selected set are performed with different scaling factor which is adapted by jDE and each mutation strategy in the pool has different scaling factor. The resulted donor vectors are then applied in crossover but all of the crossover rates are the same for each donor vector and this rate is determined by jDE as well. The results of comparison between GMDE and similar methods are illustrated in TABLE IX. As stated in this table, with respect to other methods, in majority of functions, GMDE obtained significantly better value compared to the other methods. GMDE won in 18 functions in comparison with SaDE, only lost in 6 and tied in 1 function. Moreover, in comparison with CoDE, GMDE won in 15, lost in 7 and tied in 3 functions. In the case of unimodal and multimodal functions (F1-F14), GMDE completely overcomes the other methods and achieved the best results among them in 10 functions. In the case of hybrid functions, GMDE and CoDE, both found the best mean value in 5 functions and SaDE was successful in 4 functions. Moreover, comparing to other well-known DE variants, GMDE proved to be effective and competitive as presented in TABLE X. According to this table, GMDE overcomes, DERL, Ranking-based and Proximity-based mutations which means there is significant difference between obtained average performances of GMDE and corresponding method statistically in completely fair and equal conditions. In this table, MR- indicates mean of negative ranks, MR+ indicates mean of positive ranks, SR- denotes sum of negative ranks, SR+ denotes sum of positive ranks, P-value is the measure that determines the difference between algorithms is significant or not and finally in the last column, “+” indicates that GMDE is significantly better than the compared method, “-” indicates that GMDE is significantly worse than the compared method and “=” indicates that there is no meaningful difference between GMDE and the compared method. This method holds in the context of these experiments, under benchmark functions and settings used.

#### V. CONCLUSION AND FUTURE WORKS

In this article, we proposed a new notation for DE which provides us with mutation generation and extraction capabilities. By utilizing this notation, 57 mutations were examined in preprocess phase and 16 of them were chose for second round evaluation with more function evaluation and parameter adaptation by jDE. After the second round evaluation, those mutations that were able to find the best result among the others at least in one function were chose in order to

generate mutation pools and these top 8 mutations were grouped into two mutation pools.

At each generation, only one of these pools is selected according to SSR parameter and all of the mutations in it are performed which means for each member of population 4 donor vector are generated. After crossover, the best trial vector among these 4 is chosen. The main advantage of GMDE is its synergism because of the fact that in each pool there are both explorative and exploitative mutations, hence, lack of exploitation in DE is solved. The results of extensive experiments on CEC2005 benchmarks stated that GMDE was significantly better than other methods with multiple mutations such as SaDE and CoDE and more importantly, GMDE also proved to be better than some other well-known DE variants. For future works, investigation of the effect of population size and dimensionality on this method is of great importance and significance we also intend to work on adaptive GMDE for controlling SSR.

#### ACKNOWLEDGMENT

The authors would like to thank Center of Excellence on Soft Computing and Intelligent Information Processing (SCIIP) for kind supports and Dr. Y. Wang for making the source code of jDE, SaDE and CoDE available online.

#### REFERENCES

- [1] T. Bäck, *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*: Oxford University Press, 1996.
- [2] B. Thomas, B. F. David, and M. Zbigniew, *Handbook of Evolutionary Computation*: IOP Publishing Ltd., 1997.
- [3] S. Das and P. N. Suganthan, "Differential Evolution: A Survey of the State-of-the-Art," *Evolutionary Computation, IEEE Transactions on*, vol. 15, pp. 4-31, 2011.
- [4] 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, 1997/12/01 1997.
- [5] K. Price, R. M. Storn, and J. A. Lampinen, *Differential Evolution: A Practical Approach to Global Optimization (Natural Computing Series)*: Springer-Verlag New York, Inc., 2005.
- [6] F. Neri and V. Tirronen, "Recent advances in differential evolution: a survey and experimental analysis," *Artif. Intell. Rev.*, vol. 33, pp. 61-106, 2010.
- [7] I. Boussaïd, J. Lepagnot, and P. Siarry, "A survey on optimization metaheuristics," *Information Sciences*, vol. 237, pp. 82-117, 7/10/ 2013.
- [8] W. Hui, S. Rahnamayan, S. Hui, and M. G. H. Omran, "Gaussian Bare-Bones Differential Evolution," *Cybernetics, IEEE Transactions on*, vol. 43, pp. 634-647, 2013.
- [9] N. Noman and H. Iba, "Accelerating Differential Evolution Using an Adaptive Local Search," *Evolutionary Computation, IEEE Transactions on*, vol. 12, pp. 107-125, 2008.
- [10] Matej, repin, ek, S.-H. Liu, and M. Mernik, "Exploration and exploitation in evolutionary algorithms: A survey," *ACM Comput. Surv.*, vol. 45, pp. 1-33, 2013.
- [11] R. A. Sarker, S. M. Elsayed, and T. Ray, "Differential Evolution With Dynamic Parameters Selection for Optimization Problems," *Evolutionary Computation, IEEE Transactions on*, vol. 18, pp. 689-707, 2014.
- [12] S. Das, A. Abraham, U. K. Chakraborty, and A. Konar, "Differential Evolution Using a Neighborhood-Based Mutation Operator," *Evolutionary Computation, IEEE Transactions on*, vol. 13, pp. 526-553, 2009.
- [13] G. Wenjin and C. Zhihua, "Differential Evolution With Ranking-Based Mutation Operators," *Cybernetics, IEEE Transactions on*, vol. 43, pp. 2066-2081, 2013.
- [14] E. Mezura-Montes, J. Vel, zquez-Reyes, and C. A. C. Coello, "A comparative study of differential evolution variants for global optimization," presented at the Proceedings of the 8th annual conference on Genetic and evolutionary computation, Seattle, Washington, USA, 2006.
- [15] H.-Y. Fan and J. Lampinen, "A Trigonometric Mutation Operation to Differential Evolution," *Journal of Global Optimization*, vol. 27, pp. 105-129, 2003/09/01 2003.
- [16] B. Y. Qu, P. N. Suganthan, and J. J. Liang, "Differential Evolution With Neighborhood Mutation for Multimodal Optimization," *Evolutionary Computation, IEEE Transactions on*, vol. 16, pp. 601-614, 2012.
- [17] I. Poikolainen, F. Neri, and F. Caraffini, "Cluster-Based Population Initialization for differential evolution frameworks," *Information Sciences*, vol. 297, pp. 216-235, 3/10/ 2015.
- [18] R. Mukherjee, G. R. Patra, R. Kundu, and S. Das, "Cluster-based differential evolution with Crowding Archive for niching in dynamic environments," *Information Sciences*, vol. 267, pp. 58-82, 5/20/ 2014.
- [19] Y. Zhou, X. Li, and L. Gao, "A differential evolution algorithm with intersect mutation operator," *Applied Soft Computing*, vol. 13, pp. 390-401, 1// 2013.
- [20] S. Biswas, S. Kundu, and S. Das, "An Improved Parent-Centric Mutation With Normalized Neighborhoods for Inducing Niching Behavior in Differential Evolution," *Cybernetics, IEEE Transactions on*, vol. 44, pp. 1726-1737, 2014.
- [21] Y. Wei-Jie, S. Meie, C. Wei-Neng, Z. Zhi-Hui, G. Yue-Jiao, L. Ying, *et al.*, "Differential Evolution With Two-Level Parameter Adaptation," *Cybernetics, IEEE Transactions on*, vol. 44, pp. 1080-1099, 2014.
- [22] J. Liu and J. Lampinen, "A Fuzzy Adaptive Differential Evolution Algorithm," *Soft Computing*, vol. 9, pp. 448-462, 2005/06/01 2005.
- [23] J. Brest, S. Greiner, B. Boskovic, M. Mernik, and V. Zumer, "Self-Adapting Control Parameters in Differential Evolution: A Comparative Study on Numerical Benchmark Problems," *Evolutionary Computation, IEEE Transactions on*, vol. 10, pp. 646-657, 2006.

- [24] Z. Jingqiao and A. C. Sanderson, "JADE: Adaptive Differential Evolution With Optional External Archive," *Evolutionary Computation, IEEE Transactions on*, vol. 13, pp. 945-958, 2009.
- [25] W. Yong, C. Zixing, and Z. Qingfu, "Differential Evolution With Composite Trial Vector Generation Strategies and Control Parameters," *Evolutionary Computation, IEEE Transactions on*, vol. 15, pp. 55-66, 2011.
- [26] A. K. Qin, V. L. Huang, and P. N. Suganthan, "Differential Evolution Algorithm With Strategy Adaptation for Global Numerical Optimization," *Evolutionary Computation, IEEE Transactions on*, vol. 13, pp. 398-417, 2009.
- [27] M. G. Epitropakis, D. K. Tasoulis, N. G. Pavlidis, V. P. Plagianakos, and M. N. Vrahatis, "Enhancing Differential Evolution Utilizing Proximity-Based Mutation Operators," *Evolutionary Computation, IEEE Transactions on*, vol. 15, pp. 99-119, 2011.
- [28] P. Kaelo and M. M. Ali, "A numerical study of some modified differential evolution algorithms," *European Journal of Operational Research*, vol. 169, pp. 1176-1184, 3/16/ 2006.
- [29] J. J. Liang, B. Y. Qu, X. B. Mao, B. Niu, and D. Y. Wang, "Differential evolution based on fitness Euclidean-distance ratio for multimodal optimization," *Neurocomputing*, vol. 137, pp. 252-260, 8/5/ 2014.
- [30] P. N. Suganthan, N. Hansen, J. J. Liang, Y.-P. C. K. Deb, A. Auger, and S. Tiwari, "Problem Definitions and Evaluation Criteria for the CEC2005 Special Session on Real-Parameter Optimization," *Nanyang Technol. Univ., Singapore*, 2005.





TABLE I MUTATIONS 1-9 IN PREPROCESSING PHASE FOR N=1 AT D=10 FOR 10 000 FUNCTION EVALUATION AND 50 INDEPENDENT RUN FOR F1-F16

No.	1	2	3	4	5	6	7	8	9
Xt1	rand	rand	rand	rand	rand	rand	rand	rand	rand
Xt2	rand	rand	rand	best	best	best	current	current	current
Xt3	rand	best	current	rand	best	current	rand	best	current
Func.	mean	mean	mean	mean	mean	mean	mean	mean	mean
F1	<b>4.895E-08</b>	950.34288	<b>2.756E-06</b>	<b>5.116E-15</b>	1174.7591	<b>5.386E-10</b>	3881.1771	1570.2037	666.755
F2	0.0020743	2538.9638	0.0243777	<b>1.498E-07</b>	2625.4498	<b>4.411E-05</b>	5863.1457	3708.5001	2853.5077
F3	<b>231.8239</b>	646313.77	529.48372	<b>45.996027</b>	8439833.2	<b>200.16449</b>	23864481	1399723.6	11095252
F4	<b>0.0039093</b>	3408.5968	0.088025	<b>5.234E-06</b>	2170.4329	<b>0.0017653</b>	6592.6364	4233.4267	3183.8795
F5	2.5657524	6908.1495	86.432558	<b>4.047E-05</b>	4566.2547	72.398835	13165.011	8471.7575	5511.9713
F6	2.8396309	32186660	5.7889546	<b>0.1101824</b>	42331714	<b>0.0720681</b>	890022439	224286513	48284988
F7	1267.0909	1651.3933	1270.8238	1267.0459	1618.5704	1270.8611	2478.8118	1844.694	1616.6722
F8	20.289639	20.279923	20.338206	20.256254	20.402091	20.18339	20.21062	20.298083	20.536435
F9	15.93987	15.345062	23.743943	<b>0.5792371</b>	19.516271	23.052758	34.457569	31.969457	33.258853
F10	20.292122	37.594886	28.424762	6.9653885	52.640704	29.032315	64.430588	58.151653	58.123097
F11	7.4576506	5.4570709	8.0301033	<b>0.9489766</b>	9.2003343	8.3724709	7.2644326	6.1978289	8.6599005
F12	18020.908	5130.8699	12026.721	9025.2426	6323.046	18177.241	9847.2383	9622.3717	14897.587
F13	2.1314766	2.2675844	1.6068396	<b>0.8313513</b>	5.6763293	2.1247511	1.2068078	4.2946102	6.4694046
F14	3.1627403	3.7753223	3.3540625	2.9843948	3.782158	3.0969296	3.537534	3.6902099	3.9897153
F15	123.3037	246.61309	196.93842	<b>12.23775</b>	332.76102	208.16365	409.29995	523.27001	354.75511
F16	<b>130.89825</b>	189.3308	<b>134.41044</b>	<b>117.36799</b>	251.37287	<b>55.701062</b>	271.49088	223.2765	248.8659

TABLE II MUTATIONS 1-9 IN PREPROCESSING PHASE FOR N=2 AT D=10 FOR 10 000 FUNCTION EVALUATION AND 50 INDEPENDENT RUN FOR F1-F16

No.	1	2	3	4	5	6	7	8	9
Xt1	rand	rand	rand	rand	rand	rand	rand	rand	rand
Xt2	rand	rand	rand	best	best	best	current	current	current
Xt3	rand	best	current	rand	best	current	rand	best	current
Xt4	rand	rand	rand	rand	rand	rand	rand	rand	rand
Xt5	rand	rand	rand	rand	rand	rand	rand	rand	rand
Func.	mean	mean	mean	mean	mean	mean	mean	mean	mean
F1	0.0394326	0.0247541	0.1814472	<b>0.0001488</b>	<b>4.979E-08</b>	0.0037167	21.200847	0.4500482	<b>5.398E-08</b>
F2	6.6982129	74.438793	18.579717	0.1202431	<b>0.0008023</b>	6.2293226	177.05293	617.3613	<b>0.0004582</b>
F3	65514.602	88900.081	177129.03	5254.6962	582.55828	33795.971	16469.994	614368.3	388.00996
F4	15.282998	156.35712	60.790306	0.3365822	<b>0.0021858</b>	11.754783	15.736099	152.13837	<b>0.003972</b>
F5	109.8304	1154.3885	587.02517	<b>0.085619</b>	3.3198985	268.09853	1542.9276	1998.8722	2.7793006
F6	72.459783	75.242939	294.60555	6.0409723	<b>0.7478316</b>	64.7529	59554.98	1258.571	3.5509549
F7	<b>1268.7233</b>	1341.722	1282.8061	1267.0461	1267.081	1272.4269	1419.9153	1394.694	1267.0908
F8	20.299992	20.267933	20.307757	20.291151	20.292028	20.187003	20.26842	20.236723	20.264611
F9	22.308581	29.073351	20.606688	25.507884	21.820826	23.048762	17.879867	41.832768	20.072107
F10	31.622181	44.935239	33.469539	20.584041	20.703098	28.404428	22.963164	51.716943	24.990913
F11	8.6531448	8.2887271	6.7829893	7.9540088	8.3027047	8.1584128	7.5764077	7.110081	5.7053385
F12	12197.617	10372.169	12443.884	12191.26	14467.199	13726.335	16597.947	4620.2979	15572.027
F13	2.1735285	3.290681	2.1657148	2.5338127	2.1680036	2.6800529	1.5171873	3.84256	1.6398946
F14	<b>3.5836035</b>	3.460657	3.7384236	3.4446319	3.3732502	3.5867884	3.2726597	3.7229473	3.3248434
F15	327.94127	230.5612	289.81971	234.22435	147.82368	273.46246	202.94421	280.04639	108.18065
F16	162.97591	187.35857	155.10976	166.19308	145.76074	170.91905	130.67267	195.06907	139.04854

TABLE III MUTATIONS 10-18 IN PREPROCESSING PHASE FOR N=1 AT D=10 FOR 10 000 FUNCTION EVALUATION AND 50 INDEPENDENT RUN FOR F1-F16

No.	10	11	12	13	14	15	16	17	18
Xt1	best	best	best	best	best	best	best	best	best
Xt2	rand	rand	rand	best	best	best	current	current	current
Xt3	rand	best	current	rand	best	current	rand	best	current
Func.	mean	mean	mean	mean	mean	mean	mean	mean	mean
F1	<b>5.116E-15</b>	326.54815	<b>0</b>	<b>1.421E-14</b>	256.92988	524.30095	<b>8.527E-15</b>	804.17095	96.8617
F2	<b>6.821E-15</b>	1004.6021	<b>1.194E-14</b>	<b>8.413E-14</b>	2313.9429	2069.3254	<b>6.651E-14</b>	2327.0768	3458.0678
F3	<b>106.57863</b>	2888516	<b>148.90773</b>	800529.01	26258066	490100.59	<b>39.975111</b>	4154114.3	12556657
F4	<b>0.0002049</b>	2015.1126	<b>9.768E-09</b>	6.2931737	4175.9819	3389.6442	<b>0.0004593</b>	2457.9727	4993.5028
F5	<b>7.095E-08</b>	8720.1628	248.16274	28.819042	4994.8611	6693.113	0.0219996	8881.9488	3736.5564
F6	<b>5.741E-14</b>	10697324	<b>0.6009847</b>	3.9865801	25830893	545.09339	3.9865791	17754886	34960360
F7	1267.0459	1866.0669	1275.3955	1285.6568	1347.5425	2215.2533	1267.0459	2422.1609	1359.7065
F8	20.301799	20.229214	20.270129	20.286591	20.538372	20.33853	20.241291	20.325327	20.553498
F9	12.805099	6.6284588	<b>0.23879</b>	10.269776	5.3929239	13.562347	<b>0.9800682</b>	7.3429181	6.7641347
F10	11.939499	33.165017	10.944545	70.641052	72.360945	81.649322	37.808343	42.997691	81.001524
F11	4.369617	5.9276537	2.5571801	7.4955241	9.9807293	8.3661018	5.4690302	6.9436391	10.345401
F12	535.18794	8280.1768	15373.332	226.57176	3348.0007	13434.023	6506.9407	14284.351	7914.6542
F13	0.9434962	1.4230885	<b>0.3684536</b>	1.2794416	1.6695969	<b>0.9872178</b>	<b>0.694013</b>	<b>0.8260566</b>	1.696743
F14	<b>2.0775641</b>	3.1713576	2.2905241	3.822164	3.6705872	3.6751955	2.2000774	3.6097516	3.7587218
F15	108.5704	470.56505	421.30029	426.56938	489.19619	144.91907	617.62655	214.76204	97.517107
F16	<b>152.5507</b>	214.46686	<b>159.22202</b>	173.62891	226.54162	216.33815	<b>115.32226</b>	270.48903	215.4288

TABLE IV MUTATIONS 10-18 IN PREPROCESSING PHASE FOR N=2 AT D=10 FOR 10 000 FUNCTION EVALUATION AND 50 INDEPENDENT RUN FOR F1-F16

No.	10	11	12	13	14	15	16	17	18
Xt1	best	best	best	best	best	best	best	best	best
Xt2	rand	rand	rand	best	best	best	current	current	current
Xt3	rand	best	current	rand	best	current	rand	best	current
Xt4	rand	rand	rand	rand	rand	rand	rand	rand	rand
Xt5	rand	rand	rand	rand	rand	rand	rand	rand	rand
Func.	mean	mean	mean	mean	mean	mean	mean	mean	mean
F1	<b>0</b>	<b>3.411E-15</b>	<b>1.457E-10</b>	<b>2.842E-15</b>	<b>5.684E-16</b>	<b>4.063E-12</b>	<b>5.684E-16</b>	1.8434094	<b>5.684E-16</b>
F2	<b>1.548E-11</b>	0.0110411	<b>0.0001912</b>	<b>4.065E-12</b>	<b>2.331E-14</b>	<b>3.491E-05</b>	<b>9.518E-11</b>	3.0262524	<b>4.547E-15</b>
F3	<b>10.260769</b>	1746.159	759.19416	1079.4298	<b>73.948047</b>	32843.604	<b>0.0016606</b>	828.90714	253.60148
F4	<b>2.286E-10</b>	6.9725528	<b>0.0016284</b>	<b>1.917E-07</b>	<b>0.0004431</b>	<b>0.0037871</b>	<b>1.193E-09</b>	0.086851	<b>2.541E-06</b>
F5	<b>1.413E-07</b>	10.702889	207.12838	<b>5.847E-07</b>	<b>6.466E-08</b>	303.84338	<b>2.862E-08</b>	3.70361	4.618E-05
F6	<b>1.427E-07</b>	18.051457	<b>0.0215594</b>	<b>4.94E-07</b>	<b>6.651E-14</b>	<b>0.1102164</b>	<b>0.0001178</b>	<b>0.0848894</b>	<b>1.717E-13</b>
F7	1267.0459	1267.1064	1275.7894	1267.0459	1267.0459	1273.9898	1267.0459	1269.2725	1267.2962
F8	20.238461	20.196281	20.261077	20.253459	20.244977	20.316983	20.213997	20.269811	20.171165
F9	16.880156	6.4075355	19.217127	5.0544091	10.039131	15.34268	17.247415	1.2160587	<b>7.0045067</b>
F10	25.582607	<b>6.1434226</b>	19.773033	<b>6.4418271</b>	17.909248	22.698964	28.65612	11.058469	38.803257
F11	<b>0.0062318</b>	<b>2.020586</b>	8.0401339	<b>2.2516235</b>	6.392644	<b>0.2890474</b>	2.098044	0.7575569	2.9494909
F12	17161.143	12580.121	12952.53	15936.744	2039.9164	9366.4581	11373.009	13369.254	<b>2254.59</b>
F13	<b>1.1496282</b>	<b>0.9525727</b>	1.8012985	<b>1.1930501</b>	<b>1.1024861</b>	1.9223463	1.5759111	1.0238343	<b>0.9768387</b>
F14	2.6474793	2.4294058	3.4359167	2.9089246	2.9900594	3.1024002	2.880869	2.4965507	3.0632183
F15	170.65837	93.252921	223.08833	426.31276	163.78683	<b>56.331436</b>	200.22027	<b>5.3606395</b>	428.57029
F16	138.12856	152.56792	156.08229	190.07983	156.97608	126.60092	130.94088	<b>106.35478</b>	149.51252

TABLE V MUTATIONS 19-27 IN PREPROCESSING PHASE FOR N=1 AT D=10 FOR 10 000 FUNCTION EVALUATION AND 50 INDEPENDENT RUN FOR F1-F16

No.	19	20	21	22	23	24	25	26	27
Xt1	current	current	current	current	current	current	current	current	current
Xt2	rand	rand	rand	best	best	best	current	current	current
Xt3	rand	best	current	rand	best	current	rand	best	current
Func.	mean	mean	mean	mean	mean	mean	mean	mean	mean
F1	3.8378695	4618.711	6334.3066	1.3364283	10108.445	444.61295	1425.3333	10944.831	5936.3097
F2	22.19674	9081.8949	6760.2599	2.7770554	9931.0161	2124.9447	937.35286	9593.8336	9196.2014
F3	134413.42	45566540	20908144	48895.096	46927454	4121524.7	4403900.5	38063889	85939871
F4	59.965324	7746.6817	8121.5421	21.190429	11822.644	3272.1469	1022.5929	9939.6977	11335.999
F5	411.56531	12371.594	12748.365	3.1767653	12905.801	10628.122	1317.3812	13463.925	12718.853
F6	2843.8903	568255312	459880850	1121.5578	856628800	43698251	28079061	1.174E+09	1.241E+09
F7	1270.3163	2436.2494	2536.3297	1267.0475	2520.9499	2274.3236	1267.4492	2458.8305	2611.1745
F8	20.263404	20.251949	20.163953	20.346747	20.677774	20.334831	20.378669	20.462248	20.700886
F9	24.096272	68.109695	31.251519	29.933912	76.521715	11.504241	48.415159	80.861992	89.406553
F10	33.064421	100.61538	54.610905	43.637479	92.694592	30.959112	74.836677	128.69039	117.55665
F11	7.1076029	7.2107322	7.4878398	4.7656366	11.56391	8.6643248	7.044917	10.012512	11.229009
F12	10733.583	11468.009	15785.042	10200.509	61715.54	10216.247	9234.7623	42903.782	71490.728
F13	2.8757582	7.797111	1.8998547	2.464003	20.941746	1.1832808	8.4974469	21.193342	19.254082
F14	3.5052857	3.8523778	3.5746088	3.3564499	4.2573075	3.429027	3.5166138	4.0224851	4.1760281
F15	281.99613	514.4583	457.30768	291.7225	843.33744	233.75539	351.63661	718.94026	765.53848
F16	180.56449	323.11089	261.16557	181.73132	407.48857	232.90669	265.55717	350.61075	453.04441

TABLE VI MUTATIONS 19-27 IN PREPROCESSING PHASE FOR N=2 AT D=10 FOR 10 000 FUNCTION EVALUATION AND 50 INDEPENDENT RUN FOR F1-F16

No.	19	20	21	22	23	24	25	26	27
Xt1	current	current	current	current	current	current	current	current	current
Xt2	rand	rand	rand	best	best	best	current	current	current
Xt3	rand	best	current	rand	best	current	rand	best	current
Xt4	rand	rand	rand	rand	rand	rand	rand	rand	rand
Xt5	rand	rand	rand	rand	rand	rand	rand	rand	rand
Func.	mean	mean	mean	mean	mean	mean	mean	mean	mean
F1	18.232856	279.06277	17.22593	7.7810626	4.2321426	<b>0.000848</b>	694.43771	566.71935	3.7969953
F2	120.33976	2537.805	110.3862	29.097243	17.406268	12.379136	815.00049	2179.4824	24.087463
F3	624839.15	16243538	21288.315	215019.47	210524.91	780.10624	2685399	12185067	262781.78
F4	227.06655	2246.0276	20.634887	50.917499	89.524259	<b>1.3109605</b>	943.88256	4863.9309	81.749249
F5	717.00343	7901.7305	1930.2137	27.609841	488.22615	1.7706877	1699.5475	6038.2959	342.53485
F6	21828.783	11681266	960533.83	3699.1009	2082.4411	2686.7044	2956986.2	42934441	2472.445
F7	1273.72	1839.9609	1391.5831	1267.078	1269.4941	1268.6268	1279.1216	1710.0723	1269.4822
F8	20.281889	20.21706	20.244609	20.317815	20.279691	20.242905	20.298513	20.186943	20.288644
F9	29.689931	63.07541	18.140566	29.210255	23.029298	<b>1.5179659</b>	33.743977	59.235026	30.42217
F10	33.708879	64.354727	20.703064	32.743675	38.639314	21.411825	53.926969	61.693359	35.699036
F11	8.0454814	7.4903819	7.2780147	7.5867246	7.5079605	<b>0.077874</b>	7.0985435	7.6438486	8.0080335
F12	8861.0355	11216.282	12231.931	9430.907	17060.708	11610.897	5922.3498	5983.1759	11680.151
F13	2.6196702	5.3867325	1.5714479	2.8648282	2.6812083	<b>1.0575421</b>	5.0107952	8.9465071	2.8071595
F14	3.5827826	3.7642357	3.0291006	3.4382766	3.5007625	2.8269117	3.7223785	3.6976625	3.5224865
F15	262.5709	454.78996	169.04095	285.70001	320.20538	<b>35.847088</b>	280.87753	382.96619	311.22495
F16	197.35313	257.48554	127.71187	159.8994	188.84406	<b>107.42974</b>	235.82464	244.82431	188.61859

TABLE VII SECOND ROUND EVALUATION FOR N=1 SELECTED MUTATIONS AT D=30 AND 300 000 FUNCTION EVALUATION  
FOR F1-F25 FOR 50 INDEPENDENT RUNS

prob.	DE/rand/1	DE/best/1	GMDE(rand,jDE,best,current)			GMDE(rand,jDE,best,rand)			GMDE(best,jDE,rand,current)		
	mean	mean	mean	rand/1	best/1	mean	rand/1	best/1	mean	rand/1	best/1
F1	4.04E-30	5.09E-23	<b><u>1.01E-30</u></b>	=	+	1.26E-30	=	+	1.76E-28	-	+
F2	2.75E-10	5.78E-14	2.76E-10	=	-	<b><u>6.46E-15</u></b>	+	-	1.78E-11	+	-
F3	123394.55	<b><u>20502.34</u></b>	303925.71	-	-	92778.574	+	-	114412.25	+	-
F4	0.4701517	126.96836	19.493925	-	+	<b><u>0.001642</u></b>	+	+	82.082707	-	-
F5	<b><u>1058.222</u></b>	3627.2621	2444.9438	-	+	1232.9177	-	+	4354.1001	-	-
F6	<b><u>0.419059</u></b>	1.4351846	22.385359	-	-	0.4783949	+	+	10.741449	-	-
F7	4696.2886	4696.2886	<b><u>4696.289</u></b>	=	=	4696.2886	=	=	4696.2886	=	=
F8	20.94976	20.905404	20.942895	+	-	<b><u>20.86473</u></b>	+	-	20.939895	+	-
F9	0.0198992	82.378078	<b><u>0</u></b>	+	+	0.0596975	-	+	11.859895	-	+
F10	<b><u>41.35748</u></b>	166.39893	46.251928	-	+	45.646417	-	+	126.0676	-	+
F11	26.209392	27.139876	27.788263	-	-	<b><u>23.71266</u></b>	+	+	26.74849	+	+
F12	<b><u>2162.613</u></b>	18344.462	7846.6436	-	+	7768.6152	-	+	10908.914	-	+
F13	1.2604606	5.9005635	1.3514009	-	+	<b><u>1.197619</u></b>	+	+	1.9640939	-	+
F14	12.857151	12.626637	12.908107	-	-	12.664368	+	-	<b><u>12.28676</u></b>	+	+
F15	332	451.59337	<b><u>324.0307</u></b>	+	+	337.0212	=	+	370.66586	-	+
F16	<b><u>68.7735</u></b>	251.32285	129.4695	-	+	158.36935	-	+	257.04876	-	-
F17	<b><u>132.6969</u></b>	208.04133	182.46834	-	+	200.34012	-	+	244.35722	-	-
F18	<b><u>905.457</u></b>	941.02803	906.08436	-	+	905.74803	-	+	925.91694	-	+
F19	905.58978	945.79535	906.01463	-	+	<b><u>905.4085</u></b>	+	+	926.86447	-	+
F20	<b><u>905.1445</u></b>	943.05386	905.99327	-	+	905.89374	-	+	927.91905	-	+
F21	500.00005	809.52169	<b><u>500</u></b>	=	+	518.84497	-	+	797.69293	-	+
F22	879.3692	980.67431	885.75297	-	+	<b><u>877.1381</u></b>	+	+	956.72959	-	+
F23	<b><u>534.1653</u></b>	1020.8818	544.10845	-	+	552.20802	-	+	866.97282	-	+
F24	<b><u>200</u></b>	306.09779	<b><u>200</u></b>	=	+	<b><u>200</u></b>	=	+	223.54336	=	=
F25	1625.4393	1619.2639	1622.6979	+	-	1624.0641	+	-	<b><u>1616.18</u></b>	+	+
w-t-1	-	-	-	5/3/2017	17-1-7	-	11/4/2010	19-1-5	-	6/2/2017	15-2-8

TABLE VIII SECOND ROUND EVALUATION FOR N=2 SELECTED  
MUTATIONS BY WILCOXON SIGNED-RANK TEST FOR  $\alpha=0.05$

Old Vs New	w/l/t	Old Vs New	w/l/t	Old Vs New	w/l/t
1 Vs 11	5/19/2001	1 Vs 12	7/16/2002	1 Vs 13	6/18/2001
4 Vs 11	4/20/2001	4 Vs 12	8/15/2002	4 Vs 13	6/18/2001
10 Vs 11	3/21/2001	10 Vs 12	18/6/1	10 Vs 13	12/11/2002
21 Vs 11	4/20/2001	21 Vs 12	8/15/2002	21 Vs 13	9/15/2001
24 Vs 11	4/20/1	24 Vs 12	8/16/1	24 Vs 13	7/17/1
Old Vs New	w/l/t	Old Vs New	w/l/t	Old Vs New	w/l/t
1 Vs 15	8/17/0	1 Vs 16	9/16/0	1 Vs 17	8/17/0
4 Vs 15	8/17/0	4 Vs 16	8/17/0	4 Vs 17	8/17/0
10 Vs 15	19/6/0	10 Vs 16	15/10/0	10 Vs 17	15/10/0
21 Vs 15	9/16/0	21 Vs 16	9/16/0	21 Vs 17	9/17/0
24 Vs 15	9/16/0	24 Vs 16	10/15/0	24 Vs 17	8/17/0



TABLE IX  
COMPARISON OF THE PERFORMANCE  
BETWEEN GMDE, SaDE AND CoDE  
FOR FUNCTIONS F1–F25 AT D = 30 AND  
300 000 FUNCTION EVALUATION FOR 50  
INDEPENDENT RUNS

prob.	SaDE mean	CoDE mean	GMDE mean
F1	<b>0</b>	2.71E-28	2.02E-30
F2	0.0001472	5.07E-08	<b>2.66E-27</b>
F3	560330.32	251628.52	<b>5326.7076</b>
F4	246.07652	0.02074	<b>1.44E-20</b>
F5	3247.0781	<b>48.65995</b>	203.28572
F6	44.745119	2.8220948	<b>1.5946495</b>
F7	<b>4696.2886</b>	<b>4696.2886</b>	4692.6966
F8	20.945211	20.915662	<b>20.874291</b>
F9	0.1989918	3.16E-11	<b>0</b>
F10	49.469284	36.311177	<b>29.291502</b>
F11	17.913267	<b>6.1152497</b>	23.184139
F12	3952.8558	2894.088	<b>2162.6128</b>
F13	3.8825184	5.7493565	<b>0.8128566</b>
F14	12.677943	13.055089	<b>12.148485</b>
F15	416.4883	408	<b>344.55083</b>
F16	114.20585	<b>55.446711</b>	68.773496
F17	75.429311	<b>54.866803</b>	100.50084
F18	<b>868.3646</b>	903.89464	906.30714
F19	<b>873.84459</b>	903.78894	907.79672
F20	<b>887.32372</b>	903.75403	904.65976
F21	526.23646	<b>500.00005</b>	<b>500.00005</b>
F22	935.01315	888.95499	<b>871.75524</b>
F23	534.16632	<b>534.16417</b>	534.16418
F24	<b>200</b>	<b>200</b>	<b>200</b>
F25	1645.8388	1643.3732	<b>1616.9908</b>
w/l/t	18/6/1	15/7/3	-

TABLE X RESULTS OF WILCOXON TEST AT 95% FOR F1-F25 AT  
D=30 FOR 300 000 FUNCTION EVALUATION BETWEEN GMDE,  
DERL, RANKING AND PROXIMITY VARIANTS

Algorithm	MR-	MR+	SR-	SR+	P- value	Difference
DERL Vs. GMDE	12	0	276	0	0	+
Proximity Vs. GMDE	12.56	7.6	215	38	0.002	+
Ranking Vs. GMDE	13.75	8	220	56	0.0065	+