Hybrid Differential Evolution with Convex Mutation

Jingfeng Yan[1], Chaofeng Guo[1], Wenyin Gong*[2]

[1] School of Computer Science and Technology,
Xuchang University, Xuchang, Henan, 461000, P.R. China,
Email: jeffery8224@126.com
[2] School of Computer Science,
China University of Geosciences, Wuhan 430074, P.R. China,
Email: cug11100304@yahoo.com.cn

***Abstract*— Differential evolution (DE) is a simple yet powerful evolutionary algorithm for global numerical optimization. In this paper, we propose a novel hybrid DE variant to accelerate the convergence rate of the classical DE algorithm. The proposed algorithm is hybridized with a convex mutation. The convex mutation is able to utilize the information of the parents, and hence, provides faster convergence speed. Our proposal is referred to as Convex-DE. In order to verify our expectation, we test our approach on 13 widely used benchmark functions. The results indicate that our approach is better than the classical DE algorithm in terms of the convergence speed and the quality of final solution. Furthermore, the potential of our approach for real-world applications is evaluated on three real-world problems.**

***Index Terms*— Differential evolution; convex mutation; numerical optimization; real-world applications.**

I. INTRODUCTION

Without loss of generality, a global minimization problem can be formalized as a pair (S, f) , where $S \subseteq \Re^D$ is a bounded set on $\Re^D$ and $f : S \rightarrow \Re$ is a D-dimensional real-valued function. The problem is to find a point $\mathbf{x}^* \in S$ such that $f(\mathbf{x}^*)$ is a global minimum on S [1]. More specifically, it is required to find an $\mathbf{x}^* \in S$ such that

$$\forall \mathbf{x} \in S : f(\mathbf{x}^*) \leq f(\mathbf{x}), \mathbf{x} = \{x_1, \cdots, x_i, \cdots, x_D\} \quad (1)$$

where f does not need to be continuous but it must be bounded. Generally, for each variable x_i it satisfies a constrained boundary:

$$l_i \leq x_i \leq u_i, i = 1, 2, \cdots, D \quad (2)$$

Differential Evolution (DE) [2] is a simple yet powerful population-based, direct search algorithm with the generation-and-test feature for global optimization problems using real-valued parameters. DE uses the distance and direction information from the current population to guide the further search. It won the third place at the first International Contest on Evolutionary Computation on a real-valued function test-suite [3]. Among DE's advantages are its simple structure, ease of use, speed and robustness. Price and Storn [2] gave the working principle of DE with single scheme. Later on, they suggested ten different schemes of DE [3], [4]. However, DE has been shown to have certain weaknesses, especially if the global optimum should be located using a limited number of fitness function evaluations (NFFEs). In addition, DE is good at exploring the search space and locating the region of global minimum, but it is slow exploiting of the solution [5].

In this paper, in order to accelerate the convergence speed of the classical DE algorithm, we propose a convex mutation to utilize the information of parents efficiently. The convex mutation is combined with the original DE/rand/1 mutation. The proposed approach is referred to as Convex-DE. Thirteen benchmark functions are chosen from the literature as the test suit. Our approach is compared with the classical DE algorithm with and without parameter adaptation. Experimental results show that our approach is better than the classical DE algorithm in terms of the convergence speed and the quality of final solution. In addition, three real-world problems are selected to validate the ability of our approach to solve the real-world problems.

The rest of the paper is organized as follows. Section II briefly describes the related work. Our proposed work is presented in detail in Section III, followed by the experimental results and analysis in Section IV. In the last section, Section V, we conclude our work and devote to the future work.

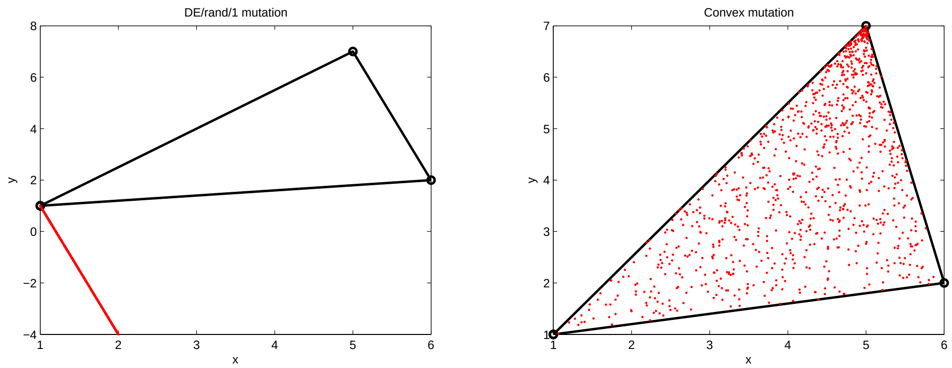
II. RELATED WORK

As mentioned above, DE is good at exploring the search space and locating the region of global minimum, but it is slow exploiting of the solution [5]. Recently, many researchers are working on the improvement of DE hybridized with other methods. Fan and Lampinen [6]

This work is supported by the Foundation for Young Core Teachers of Higher Education Institutions in Henan Province under Grant No. 2009GGJS-120.

* Corresponding author.

doi:10.4304/jsw.6.11.2321-2328



(a) (b)

Figure 1. Illustration of DE/rand/1 mutation and convex mutation in $2D$ search space. (a) DE/rand/1 mutation; (b) Convex mutation.

proposed a new version of DE which uses an additional mutation operation called trigonometric mutation operation. They showed that the modified DE algorithm can outperform the classic DE algorithm for some benchmarks and real-world problems. Sun *et al.* [7] proposed a new hybrid algorithm based on a combination of DE and Estimation of Distribution Algorithm (EDA). This technique uses a probability model to determine promising regions in order to focus the search process on those areas. Gong *et al.* [8] employed the two level orthogonal crossover to improve the performance of DE. They showed that the proposed approach performs better than the classical DE in terms of the quality, speed, and stability of the final solutions. Noman and Iba [9] proposed fittest individual refinement, a crossover-based local search (LS) method DE to solve the high dimensional problems. They showed that the improved DE method accelerates the convergence rate for high dimensional benchmark functions. Based on their previous work, Noman and Iba incorporated LS into the classical DE in [5]. They presented an LS technique to solve this problem by adaptively adjusting the length of the search, using a hill-climbing heuristic. Through the experiments, they showed that the proposed new version of DE performs better, or at least comparably, to classic DE algorithm. Kaelo and Ali [10] adopted the attraction-repulsion concept of electromagnetism-like algorithm to boost the mutation operation of the original DE. Yang *et al.* [11] proposed a neighborhood search based DE. Experimental results showed that DE with neighborhood search has significant advantages over other existing algorithms on a broad range of different benchmark functions [11]. Wang *et al.* [12] proposed a dynamic clustering-based DE for global optimization, where a hierarchical clustering method is dynamically incorporated in DE. Experiments on 28 benchmark problems, including 13 high dimensional functions, showed that the new method is able to find near optimal solutions efficiently [12]. Recently, Cai and Gong [13] presented a clustering-based DE variant, where the one-step K-means algorithm is used to improve the performance of the original DE algorithm.

III. OUR APPROACH: CONVEX-DE

In this section, we first point out the motivations of this work. Then, the convex mutation operation is presented, followed by the flowchart of our proposed Convex-DE method.

A. Motivations

Differential evolution is a simple and robust global optimization method, which has shown powerful ability in many different applications. In DE, the main operation is the *differential mutation*, which is the core operation of DE. The classical differential mutation is "DE/rand/1" as follows:

$$\mathbf{v}_i = \mathbf{x}_{r_1} + F(\mathbf{x}_{r_2} - \mathbf{x}_{r_3}) \tag{3}$$

where $i, r_1, r_2, r_3 \in [1, D]$, $i \neq r_1 \neq r_2 \neq r_3$, D is the dimension of decision variables. $F \in (0, 1]$ is the scaling factor. $\mathbf{v}_i$ is the mutant vector. Suppose we have 3 points in $2D$ search space, they are $\mathbf{x}_{r_1} = (1, 1)$,$\mathbf{x}_{r_2} = (6, 2)$, and $\mathbf{x}_{r_3} = (5, 7)$ as shown in Fig. 1 (a) in "$\circ$". The three points form a triangle. We generate $1,000$ points with different F values using Eqn. (3), and the figure is plotted in Fig. 1 (a). The possible mutants are aligned from $(1, 1)$ to $(2, -4)$.

The "DE/rand/1" mutation can be reformulated as follows:

$$\mathbf{v}_i = a_1\mathbf{x}_{r_1} + a_2\mathbf{x}_{r_2} + a_3\mathbf{x}_{r_3} \tag{4}$$

where $a_1 = 1.0, a_2 = F, a_3 = -F$, and $a_1 + a_2 + a_3 = 1.0$. As shown in Fig. 1 (a), "DE/rand/1" mutation performs the non-convex search. The mutants are only aligned a line. Therefore, "DE/rand/1" mutation is not able to use the information of parents efficiently.

B. Convex Mutation

In order to utilize the information of parents more efficiently, we propose a *convex* mutation. The convex mutation is shown in Eqn. (4). In the convex mutation, the coefficients $a_i \in (0, 1)$, $i = 1, 2, 3$, and $\sum_{i=1}^{3} a_i = 1.0$. Since all the coefficients $a_i \in (0, 1)$, Eqn. (4) performs the convex search. Suppose we also have three points

TABLE I.
THE 13 BENCHMARK FUNCTIONS USED IN OUR EXPERIMENTAL STUDY, WHERE D IS THE NUMBER OF VARIABLES AND $S \subseteq \mathbb{R}^D$. EACH OF THEM HAS A GLOBAL MINIMUM VALUE OF 0. A DETAIL DESCRIPTION OF ALL FUNCTIONS CAN BE FOUND IN [1].

Name	Test Functions	S
Sphere	$f_{01} = \sum_{i=1}^{D} x_i^2$	$[-100, 100]^D$
Schwefel 2.22	$f_{02} = \sum_{i=1}^{D} \lvert x_i\rvert + \prod_{i=1}^{D} \lvert x_i\rvert$	$[-10, 10]^D$
Schwefel 1.2	$f_{03} = \sum_{i=1}^{D} (\sum_{j=1}^{i} x_j)^2$	$[-100, 100]^D$
Schwefel 2.21	$f_{04} = \max_i\{\lvert x_i\rvert, 1 \le i \le D\}$	$[-100, 100]^D$
Rosenbrock	$f_{05} = \sum_{i=1}^{D-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$[-30, 30]^D$
Step	$f_{06} = \sum_{i=1}^{D-1} (\lfloor x_i + 0.5\rfloor)^2$	$[-100, 100]^D$
Quartic	$f_{07} = \sum_{i=1}^{D} x_i^4 + random[0, 1)$	$[-1.28, 1.28]^D$
Schwefel 2.26	$f_{08} = \sum_{i=1}^{D} (-x_i \sin(\sqrt{\lvert x_i\rvert})) + 418.98288727243369 \times D$	$[-500, 500]^D$
Rastrigin	$f_{09} = \sum_{i=1}^{D} (x_i^2 - 10\cos(2\pi x_i) + 10)$	$[-5.12, 5.12]^D$
Ackley	$f_{10} = -20\exp(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}) - \exp(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i)) + 20 + \exp(1)$	$[-32, 32]^D$
Griewank	$f_{11} = \frac{1}{4000}\sum_{i=1}^{D} x_i^2 - \prod_{i=1}^{D}\cos(\frac{x_i}{\sqrt{i}}) + 1$	$[-600, 600]^D$
Penalized 1	$f_{12} = \frac{\pi}{D}\{10\sin^2(\pi y_1) + \sum_{i=1}^{D-1}(y_i - 1)^2 \cdot [1 + 10\sin^2(\pi y_{i+1})] + (y_D - 1)^2\} + \sum_{i=1}^{D} u(x_i, 10, 100, 4)$	$[-50, 50]^D$
Penalized 2	$f_{13} = \frac{1}{10}\{\sin^2(3\pi x_1) + \sum_{i=1}^{D-1}(x_i - 1)^2[1 + \sin^2(3\pi x_{i+1})] + (x_D - 1)^2[1 + \sin^2(2\pi x_D)]\} + \sum_{i=1}^{D} u(x_i, 5, 100, 4)$	$[-50, 50]^D$

TABLE II.
COMPARISON ON THE RESULTS BETWEEN THE CLASSICAL DE WITH CONVEX-DE FOR ALL FUNCTIONS AT $D = 30$. THE BEST RESULTS ARE HIGHLIGHTED IN **BOLDFACE**.

F	Error						SP		AR'
	DE			Convex-DE			DE	Convex-DE	
	Mean	Std	S_r	Mean	Std	S_r			
f_{01}	6.41E-32†	8.33E-32	1.00	**1.58E-68**	**3.59E-68**	1.00	1.05E+05	**4.99E+04**	2.10
f_{02}	6.50E-16†	4.67E-16	1.00	**6.84E-39**	**5.93E-39**	1.00	1.76E+05	**7.37E+04**	2.39
f_{03}	2.49E-05†	2.07E-05	0.00	**8.95E-10**	**1.66E-09**	1.00	NA	**2.62E+05**	NA
f_{04}	**8.82E-02**‡	**2.19E-01**	0.00	4.05E+00	2.20E+00	0.00	NA	NA	NA
f_{05}	**1.43E+00**‡	**1.01E+00**	0.00	2.01E+01	7.49E+00	0.00	NA	NA	NA
f_{06}	0.00E+00	0.00E+00	1.00	0.00E+00	0.00E+00	1.00	3.83E+04	**1.78E+04**	2.15
f_{07}	4.71E-03†	1.21E-03	1.00	**1.70E-03**	**6.89E-04**	1.00	1.45E+05	**4.50E+04**	3.22
f_{08}	6.59E+03†	7.04E+02	0.00	**4.09E+02**	**2.80E+02**	0.04	NA	**6.85E+06**	NA
f_{09}	1.41E+02†	2.06E+01	0.00	**9.45E+00**	**3.77E+00**	0.00	NA	NA	NA
f_{10}	4.14E-15†	0.00E+00	1.00	**1.87E-15**	**1.72E-15**	1.00	1.63E+05	**7.76E+04**	2.09
f_{11}	**1.48E-04**	**1.05E-03**	0.98	4.43E-04	2.21E-03	0.96	1.11E+05	**5.39E+04**	2.06
f_{12}	1.93E-32†	6.70E-33	1.00	**1.57E-32**	**0.00E+00**	1.00	9.62E+04	**4.20E+04**	2.29
f_{13}	**1.44E-30**†	**1.80E-30**	1.00	2.51E-02	1.73E-01	0.92	1.14E+05	**5.63E+04**	2.03
$w/t/l$	9/2/2			−			−	−	−

† indicates Convex-DE is significantly better than DE by the Wilcoxon signed-rank test at $\alpha = 0.05$.
‡ means that Convex-DE is significantly worse than DE by the Wilcoxon signed-rank test at $\alpha = 0.05$.

$\mathbf{x}_{r_1} = (1, 1)$,$\mathbf{x}_{r_2} = (6, 2)$, and $\mathbf{x}_{r_3} = (5, 7)$. We generate $1,000$ points using different a_i values with Eqn. (4). The generated mutants are shown in Fig. 1 (b). From Fig. 1 (b), we can see that all generated points are in the triangle. In addition, the generated points are scattered uniformly within the triangle. Thus, the convex mutation is able to efficiently utilize the information of parents and generate more efficient mutants.

C. Convex-DE

By combing the convex mutation with "DE/rand/1" in the DE framework, we obtain our proposed Convex-DE approach. Our approach is shown in Algorithm 1. Where D is the number of decision variables. NP is the size of the parent population P. F is the mutation scaling factor. CR is the probability of crossover operator. $x_{i,j}$ is the j-th variable of the solution $\mathbf{x}_i$. $\mathbf{u}_i$ is the offspring. $\text{rndint}(1, D)$ is a uniformly distributed random integer number between 1 and n. $\text{rndreal}_j[0, 1)$ is a uniformly

Algorithm 1 Our proposed Convex-DE algorithm

```
Generate the initial population P
Evaluate the fitness for each individual in P
while The halting criterion is not satisfied do
    for i = 1 to NP do
        Select uniform randomly r1 ≠ r2 ≠ r3 ≠ i
        if rndreal[0, 1) < p_cm then
            v_i = a1 x_r1 + a2 x_r2 + a3 x_r3 {Convex mutation}
        else
            v_i = x_r1 + F × (x_r2 − x_r3) {"DE/rand/1" mutation}
        end if
        j_rand = rndint(1, D)
        for j = 1 to D do
            if rndreal_j[0, 1) < CR or j == j_rand then
                u_i,j = v_i,j
            else
                u_i,j = x_i,j
            end if
        end for
    end for
    for i = 1 to NP do
        Evaluate the offspring u_i
        if u_i is better than x_i then
            x_i = u_i
        end if
    end for
end while
```

TABLE III.
COMPARISON ON THE RESULTS BETWEEN THE DE WITH CONVEX-DE FOR ALL FUNCTIONS AT $D = 100$. THE BEST RESULTS ARE HIGHLIGHTED IN **BOLDFACE**.

F	Error						SP		AR'
	jDE			Convex-jDE			jDE	Convex-jDE	
	Mean	Std	S_r	Mean	Std	S_r			
f_{01}	1.45E+00†	3.34E-01	0.00	**3.94E-23**	**3.00E-23**	1.00	NA	**4.57E+05**	NA
f_{02}	2.80E+00†	6.30E-01	0.00	**6.09E-14**	**2.05E-14**	1.00	NA	**6.59E+05**	NA
f_{03}	1.68E+05†	1.85E+04	0.00	**8.87E+01**	**2.03E+01**	0.00	NA	NA	NA
f_{04}	1.97E+01†	2.22E+00	0.00	**6.09E+00**	**9.55E-01**	0.00	NA	NA	NA
f_{05}	2.80E+02†	5.07E+01	0.00	**1.13E+02**	**3.37E+01**	0.00	NA	NA	NA
f_{06}	2.80E-01†	8.82E-01	0.86	**0.00E+00**	**0.00E+00**	1.00	1.12E+06	**1.79E+05**	6.29
f_{07}	7.58E-02†	1.32E-02	0.00	**9.79E-03**	**1.87E-03**	0.64	NA	**8.35E+05**	NA
f_{08}	3.21E+04†	5.21E+02	0.00	**2.78E+04**	**6.18E+02**	0.00	NA	NA	NA
f_{09}	8.75E+02†	2.25E+01	0.00	**7.56E+00**	**2.70E+00**	0.00	NA	NA	NA
f_{10}	3.20E-01†	5.52E-02	0.00	**1.19E-12**	**6.45E-13**	1.00	NA	**7.03E+05**	NA
f_{11}	5.81E-01†	9.55E-02	0.00	**9.36E-04**	**3.10E-03**	0.90	NA	**5.01E+05**	NA
f_{12}	2.48E-01†	1.27E-01	0.00	**1.87E-03**	**7.46E-03**	0.94	NA	**3.82E+05**	NA
f_{13}	1.99E+01†	6.85E+00	0.00	**5.33E-03**	**3.15E-02**	0.90	NA	**5.25E+05**	NA
$w/t/l$	13/0/0			–			–	–	–

† indicates Convex-DE is significantly better than DE by the Wilcoxon signed-rank test at $\alpha = 0.05$.

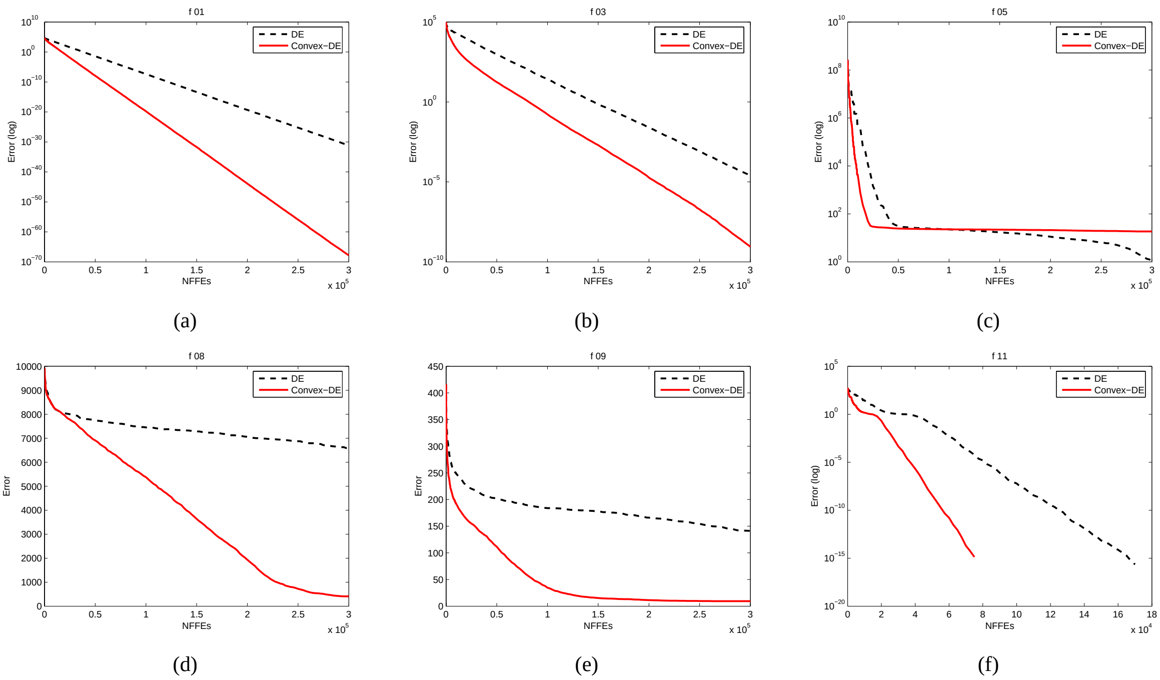


Figure 2. Convergence curves of DE and Convex-DE on the selected functions at $D = 30$. (a) f_{01}; (b) f_{03}; (c) f_{05}; (d) f_{08}; (e) f_{09}; (f) f_{11}.

distributed random real number in $[0, 1)$. $p_{cm} \in [0, 1]$ is the control parameter to control the selection between convex mutation and “DE/rand/1” mutation.

From Algorithm 1 we can see that our approach is also very simple like the classical DE algorithm. In Convex-DE, only one additional operation, convex mutation, is integrated into DE. Thus, the overall complexity of Convex-DE is the same to DE. Our approach is also very easy to implement.

IV. EXPERIMENTAL RESULTS

In this work, we have carried out different experiments using a test suite, consisting of 13 unconstrained single-objective benchmark functions with different characteristics chosen from the literature. All of the functions are minimization and scalable problems. The 13 functions, $f_{01} - f_{13}$, are chosen from [1]. The brief description of these functions are shown in Table I, more details about them can be found in [1].

Functions $f_{01} - f_{04}$ are unimodal. The generalized Rosenbrock’s function f_{05} is a multi-modal function when $D > 3$ [14]. Function f_{06} is the step function, which has one minimum and is discontinuous. Function f_{07} is a noisy quartic function. Functions $f_{08} - f_{13}$ are multi-modal functions where the number of local minima increases exponentially with the problem dimension. They appear to be the most difficult class of problems for many optimization algorithms.

TABLE IV.
COMPARISON ON THE RESULTS BETWEEN THE jDE WITH CONVEX-jDE FOR ALL FUNCTIONS AT $D = 30$. THE BEST RESULTS ARE HIGHLIGHTED IN **BOLDFACE**.

F	Error						SP		AR'
	jDE			Convex-jDE			jDE	Convex-jDE	
	Mean	Std	S_r	Mean	Std	S_r			
f_{01}	1.64E-61†	2.28E-61	1.00	**2.66E-78**	**2.90E-78**	1.00	5.89E+04	**4.65E+04**	1.27
f_{02}	1.96E-36†	1.60E-36	1.00	**1.04E-43**	**1.41E-43**	1.00	8.09E+04	**6.75E+04**	1.20
f_{03}	2.14E-06†	2.09E-06	0.00	**3.41E-07**	**6.78E-07**	0.04	NA	**7.44E+06**	NA
f_{04}	**5.38E-09**†	**4.38E-09**	0.94	1.47E-06	1.04E-05	0.86	3.07E+05	**2.98E+05**	1.03
f_{05}	8.79E+00†	1.84E+00	0.00	**8.05E+00**	**1.88E+00**	0.00	NA	NA	NA
f_{06}	0.00E+00	0.00E+00	1.00	0.00E+00	0.00E+00	1.00	2.21E+04	**1.72E+04**	1.28
f_{07}	3.50E-03†	7.54E-04	1.00	**1.56E-03**	**5.20E-04**	1.00	1.09E+05	**5.12E+04**	2.14
f_{08}	0.00E+00	0.00E+00	1.00	0.00E+00	0.00E+00	1.00	9.03E+04	**7.95E+04**	1.14
f_{09}	0.00E+00	0.00E+00	1.00	0.00E+00	0.00E+00	1.00	1.17E+05	**1.00E+05**	1.17
f_{10}	4.14E-15	0.00E+00	1.00	**3.86E-15**	**9.74E-16**	1.00	8.93E+04	**7.11E+04**	1.26
f_{11}	0.00E+00	0.00E+00	1.00	0.00E+00	0.00E+00	1.00	6.19E+04	**4.84E+04**	1.28
f_{12}	1.57E-32	0.00E+00	1.00	1.57E-32	0.00E+00	1.00	5.33E+04	**4.07E+04**	1.31
f_{13}	1.35E-32	0.00E+00	1.00	1.35E-32	0.00E+00	1.00	6.41E+04	**4.87E+04**	1.32
$w/t/l$	6/7/0			−			−	−	−

† indicates Convex-jDE is significantly better than jDE by the Wilcoxon signed-rank test at $\alpha = 0.05$.

TABLE V.
COMPARISON ON THE RESULTS BETWEEN THE DE WITH CONVEX-DE FOR REAL-WORLD PROBLEMS. THE BEST RESULTS ARE HIGHLIGHTED IN **BOLDFACE**. WHERE SEVERAL ALGORITHMS CAN OBTAIN THE GLOBAL OPTIMUM FOR A PROBLEM, THE *Intermediate* RESULTS ARE ALSO REPORTED HEREIN.

F	NFFEs	Error						SP		AR
		DE			Convex-DE			DE	Convex-DE	
		Mean	Std	Sr	Mean	Std	Sr			
CP	50000 200000	1.03E-03† 0.00E+00	8.84E-04 0.00E+00	1.00	**4.69E-05** 0.00E+00	**1.26E-04** 0.00E+00	1.00	6.29E+04	**5.17E+04**	1.22
FM	50000 200000	3.73E+00† 2.46E-01	5.68E+00 1.74E+00	0.98	**6.67E-01** 0.00E+00	**3.25E+00** 0.00E+00	1.00	6.03E+04	**4.75E+04**	1.27
LES	200000	1.19E-13†	8.76E-14	1.00	**0.00E+00**	**0.00E+00**	1.00	1.37E+05	**9.57E+04**	1.43

† indicates Convex-DE is significantly better than DE by the Wilcoxon signed-rank test at $\alpha = 0.05$.

A. Experimental Settings

In this work, for DE and Convex-DE, we have chosen a reasonable set of value and have not made any effort in finding the best parameter settings. For all experiments, we use the following parameters unless a change is mentioned.

- Dimension of each function: $D = 30$;
- Population size: $NP = 100$;
- Crossover rate: $CR = 0.9$;
- Scaling factor: $F = 0.5$;
- Convex mutation parameter: $p_{cm} = 0.05$;
- Value to reach: VTR = 10^{-8}, except for f_{07} of VTR = 10^{-2};
- Maximum number of fitness function evaluations: Max_NFFEs = $D \times 10000 = 300,000$.

In our experiments, each function is optimized over 50 independent runs. We also use the same set of initial random populations to evaluate different algorithms in a similar way done in [5]. All the algorithms are implemented in standard C++.

B. Performance Criteria

Five performance criteria are selected from the literature [15], [16], [17] to evaluate the performance of the algorithms. These criteria are described as follows.

- **Error** [15]: The error of a solution $\mathbf{x}$ is defined as $f(\mathbf{x}) - f(\mathbf{x}^*)$, where $\mathbf{x}^*$ is the global minimum of the function. The minimum error is recorded when the Max_NFFEs is reached in 50 runs. The average and standard deviation of the error values are calculated as well.
- **Successful rate (S_r)** [15]: The number of successful runs is recorded when the VTR is reached before the Max_NFFEs condition terminates the trial. Thus, the successful rate S_r is calculated as the number of successful runs divided by the total number of runs.
- **Successful performance (SP)** [17]: The number of fitness function evaluations (NFFEs) is recorded when the VTR is reached. Therefore, the successful performance is calculated as: $SP = \frac{NFFEs}{S_r}$.
- **Convergence graphs** [15]: The convergence graphs show the *median* error performance of the best solution over the total runs, in the respective experiments.
- **Acceleration rate (AR')**: In [16], the AR is presented. In this work, this measure is modified as: $AR' = \frac{\mathrm{AP_{other}}}{\mathrm{AP_{ours}}}$, where $AR' > 1$ indicates our approach is faster than its competitor.

C. General Performance

In this section, we compare the performance of Convex-DE with that of DE on all test functions. Each function is conducted over 50 independent runs. The results are shown in Table II. The convergence graphs of some selected functions are plotted in Fig. 2. In Table II, the paired Wilcoxon signed-rank test at $\alpha = 0.05$ is adopted to compare the significance between two algorithms. In the last row of this table, according to the Wilcoxon's test,

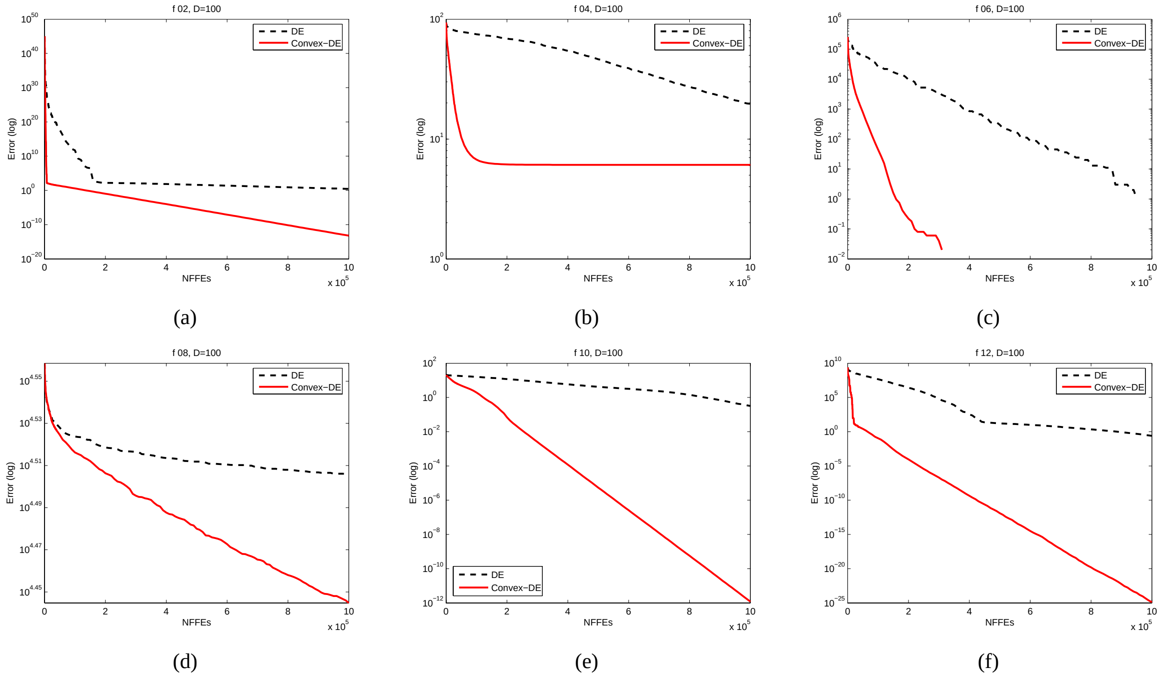


Figure 3. Convergence curves of DE and Convex-DE on the selected functions at $D = 100$. (a) f_{02}; (b) f_{04}; (c) f_{06}; (d) f_{08}; (e) f_{10}; (f) f_{12}.

the results are summarized as "$w/t/l$", which means that Convex-DE wins in w functions, ties in t functions, and loses in l functions, compared with its competitors.

With respect to the error values shown in Table II, it can be seen that Convex-DE is significantly better than DE on 9 out of 13 functions. On two functions f_{04} and f_{05}, our approach is significantly outperformed by DE. On the rest two functions (f_{06}, f_{11}), both algorithms obtain similar error values. Considering the SP measure, we can see that for all successful functions, Convex-DE obtains higher SP performance than DE. The average AR' value is greater than 2.0, which means that Convex-DE is over twice faster than DE in terms of convergence speed.

From Fig. 2, we can also find that on the majority of the test functions Convex-DE converges faster than DE. On functions f_{04} and f_{05}, Convex-DE provides faster convergence rate than DE at the beginning of evolution process. However, Convex-DE converges early on these two functions after some generations. The reason might be the fast lost of diversity of population.

D. Performance Analysis for Moderate-Dimensional Problems

In the previous section, the dimensionality of all function are set at $D = 30$, in this section, the performance of our approach is analyzed for moderate-dimensional problems at $D = 100$. As stated in [4], for higher dimensional problems, the larger population size is required in DE. Therefore, the population size $NP = 400$ for DE and Convex-DE is used. The Max_NFFEs are set as $D \times 1000 = 1000000$. All other parameters are the same as shown in Section IV-A. The results are tabulated in Table III. The convergence graph of the selected functions are shown in Fig. 3.

According to the results shown in Table III, it can be seen that when the dimensionality of the functions increases, the overall successful rates for both DE and Convex-DE decrease. However, our approach is still able to obtain higher overall successful rates compared with DE. With respect to the error values, Convex-DE significantly outperforms DE for all test 13 functions. In addition, Convex-DE obtains better successful performance than DE. From Fig. 3, we can see that Convex-DE converges faster than DE on all functions.

E. Influence of Parameter Adaptation

Since the parameter settings of DE (*i.e.*, CR and F) are sensitive to its performance [18], it is valuable to verify the influence of parameter adaptation to the proposed variant. In this section, the parameter adaptation proposed in jDE [18] is adopted in DE and Convex-DE. The two variants are referred to as jDE and Convex-jDE, respectively. The parameters in parameter adaptation are set as the same to jDE [18]. All other parameters are kept unchanged as described in Section IV-A. The results are presented in Table IV, and the convergence graphs are shown in Fig. 4.

From Table IV we can see that on 6 out of 13 functions, Convex-DE is significantly better than DE. For the rest 7 functions (f_{06}, $f_{08} - f_{13}$), both Convex-DE and DE obtains the near global optimal values on these functions. However, when considering the SP and AR' performance, Convex-DE is better than DE on all successful functions, including functions f_{06}, $f_{08} - f_{13}$. Moreover, from Fig. 4,

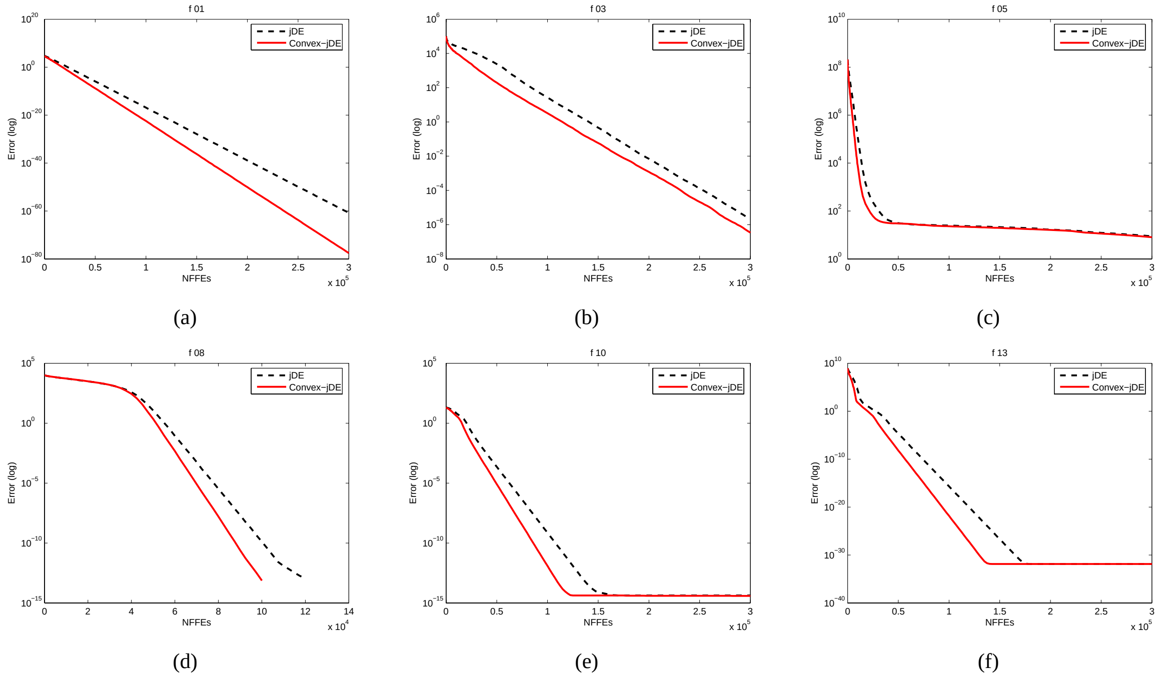


Figure 4. Convergence curves of jDE and Convex-jDE on the selected functions at $D = 100$. (a) f_{01}; (b) f_{03}; (c) f_{05}; (d) f_{08}; (e) f_{10}; (e) f_{13}.

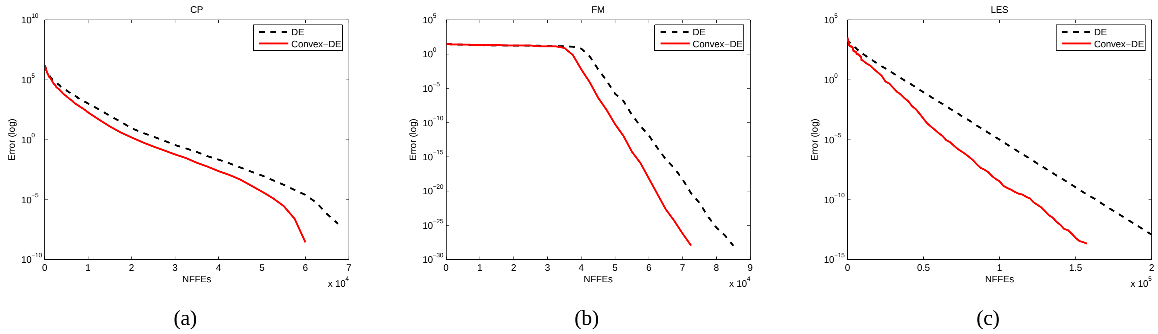


Figure 5. Convergence curves of DE and Convex-DE on the real-world problems. (a) CP problem; (b) FM problem; (c) LES problem.

we can see that Convex-DE converges faster than DE on all functions.

In general, from the results and analysis we can conclude that the parameter adaptation in [18] does not influence the improvement of our proposed Convex-DE method. Some other parameter adaptation techniques in the DE literature maybe also be benefit from the convex mutation proposed in this work.

F. Comparison on Real-World Problems

In this section, three real-world problems widely used in evolutionary algorithms are selected to test the ability of our approach for solving the real-world problems. The three problems are (i) the Chebychev polynomial fitting problem (CP) [4], (ii) the Frequency-Modulated sound waves (FM) [19], and (iii) the systems of linear equation problem (LES) [20]. The CP is defined at $D = 9$, FM problem is defined at $D = 6$, and LES at $D = 10$. For each problem, the Max_NFFEs for the three problems are set as 200000. All other parameters are kept unchanged as described in Section IV-A. The results are shown in Table V and Fig. 5. In Table V, when several algorithms can obtain the global optimum in many runs for a problem, the intermediate results are also reported for this problem. From the results, we can see that for the three real-world problems Convex-DE consistently obtains the significantly better results than DE in terms of the error values and the convergence speed.

V. Conclusions and Future Work

In this paper, a convex mutation is presented to utilize the information of parents efficiently. The convex mutation is combined with “DE/rand/1” mutation to form the hybrid DE variant, Convex-DE. Our proposed approach is also very simple and easy to implement. The experimental results on benchmark problems and three real-world problems demonstrate the superiority of our approach.

In Convex-DE, an additional parameter p_{cm} is used. In this work, the parameter set fixed. In the future work, the parameter adaptation on this parameter will be studied to further improve the performance of DE. In addition, Convex-DE may lead to premature convergence due to the fast lost of diversity of population. Thus, another direction is using the population restart to improve the performance of Convex-DE.

ACKNOWLEDGMENT

The authors would like to thank Prof. J. Brest for providing the jDE source code.

REFERENCES

[1] X. Yao, Y. Liu, and G. Lin, "Evolutionary programming made faster," *IEEE Trans. on Evol. Comput.*, vol. 3, no. 2, pp. 82–102, Jul 1999.

[2] R. Storn and K. Price, "Differential evolution–A simple and efficient heuristic for global optimization over continuous spaces," *J. of Global Optim.*, vol. 11, no. 4, pp. 341–359, Dec 1997.

[3] ——, "Home page of differential evolution," 2010. [Online]. Available: http://www.ICSI.Berkeley.edu/˜storn/code.html

[4] K. Price, R. Storn, and J. Lampinen, *Differential Evolution: A Practical Approach to Global Optimization*. Berlin: Springer-Verlag, 2005.

[5] N. Noman and H. Iba, "Accelerating differential evolution using an adaptive local search," *IEEE Trans. on Evol. Comput.*, vol. 12, no. 1, pp. 107–125, Feb 2008.

[6] H.-Y. Fan and J. Lampinen, "A trigonometric mutation operation to differential evolution," *J. of Global Optimization*, vol. 27, no. 1, pp. 105–129, 2003.

[7] J. Sun, Q. Zhang, and E. P. K. Tsang, "DE/EDA: A new evolutionary algorithm for global optimization," *Information Sciences*, vol. 169, no. 3-4, pp. 249–262, 2005.

[8] W. Gong, Z. Cai, and C. X. Ling, "ODE: A fast and robust differential evolution based on orthogonal design," in *AI 2006: Advances in Artificial Intelligence, 19th Australian Joint Conference on Artificial Intelligence, Hobart, Australia, December 4-8, 2006, Proceedings*, vol. 4304, Berlin, German, 2006, pp. 709–718.

[9] N. Noman and H. Iba, "Enhancing differential evolution performance with local search for high dimensional function optimization," in *Proc. Genetic Evol. Comput. Conf.*, H.-G. Beyer and U.-M. O'Reilly, Eds., June 25-29, 2005, pp. 967–974.

[10] P. Kaelo and M. M. Ali, "Differential evolution algorithms using hybrid mutation," *Comput. Optim. Appl.*, vol. 37, no. 2, pp. 231–246, 2007.

[11] Z. Yang, K. Tang, and X. Yao, "Self-adaptive differential evolution with neighborhood search," in *Proc. IEEE Congress on Evol. Comput.*, 2008, pp. 1110–1116.

[12] Y.-J. Wang, J.-S. Zhang, and G.-Y. Zhang, "A dynamic clustering based differential evolution algorithm for global optimization," *European Journal of Operational Research*, vol. 183, no. 1, pp. 56–73, 2007.

[13] Z. Cai, W. Gong, C. X. Ling, and H. Zhang, "A clustering-based differential evolution for global optimization," *Applied Soft Computing*, vol. 11, no. 1, pp. 1363 – 1379, January 2011.

[14] Y.-W. Shang and Y.-H. Qiu, "A note on the extended rosenbrock function," *Evol. Comput.*, vol. 14, no. 1, pp. 119–126, 2006.

[15] P. N. Suganthan, N. Hansen, J. J. Liang, K. Deb, Y.-P. Chen, A. Auger, and S. Tiwari, "Problem definitions and evaluation criteria for the CEC2005 special session on real-parameter optimization," 2005. [Online]. Available: http://www.ntu.edu.sg/home/EPNSugan

[16] S. Rahnamayan, H. Tizhoosh, and M. Salama, "Opposition-based differential evolution," *IEEE Trans. on Evol. Comput.*, vol. 12, no. 1, pp. 64–79, Feb 2008.

[17] K. Price and J. Ronkkonen, "Comparing the uni-modal scaling performance of global and local selection in a mutation-only differential evolution algorithm," in *IEEE Congr. on Evol. Comput.*, 2006, pp. 2034–2041.

[18] J. Brest, S. Greiner, B. Bošković, M. Mernik, and V. Žumer, "Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems," *IEEE Trans. on Evol. Comput.*, vol. 10, no. 6, pp. 646–657, Dec 2006.

[19] S. Das, A. Abraham, U. K. Chakraborty, and A. Konar, "Differential evolution using a neighborhood-based mutation operator," *IEEE Trans. on Evol. Comput.*, vol. 13, no. 3, pp. 526–553, 2009.

[20] F. Herrera and M. Lozano, "Gradual distributed real-coded genetic algorithms," *IEEE Trans. on Evol. Comput.*, vol. 4, no. 1, pp. 43–63, 2000.

Jingfeng Yan received the B. Eng. in computer science from Hubei University of Technology and M. Eng. in computer science from China University of Geosciences, Wuhan, in 2004 and 2007, respectively.

He is currently with Xuchang University. He has published over 10 research papers in journals and international conferences. His research interests include differential evolution, Network Security.

Chaofeng Guo received the B. Eng. in computer science from Henan Normal University and M. Eng. in computer science from Henan University, in 1997 and 2007, respectively.

He is currently an Associate Professor with Xuchang University. He has published over 15 research papers in journals and international conferences. His research interests include differential evolution, image processing.

Wenyin Gong received the B.Eng., M.Eng., and Ph.D. degrees in computer science from China University of Geosciences, Wuhan, China, in 2004, 2007, and 2010, respectively.

He is currently with China University of Geosciences. He has published over 30 research papers in journals and international conferences. His research interests include differential evolution, memetic algorithms, multiobjective optimization, and their applications.