

Research Article

Sufficient Conditions for Global Convergence of Differential Evolution Algorithm

Zhongbo Hu,^{1,2} Shengwu Xiong,¹ Qinghua Su,² and Xiaowei Zhang³

¹ School of Computer Science and Technology, Wuhan University of Technology, Wuhan, Hubei 430070, China

² School of Mathematics and Statistics, Hubei Engineering University, Xiaogan, Hubei 432100, China

³ School of Mathematical Sciences, University of Electronic Science and Technology of China, Chengdu, Sichuan 611731, China

Correspondence should be addressed to Zhongbo Hu; huzbdd@126.com

Received 6 July 2013; Revised 11 August 2013; Accepted 27 August 2013

Academic Editor: Yongkun Li

Copyright © 2013 Zhongbo Hu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The differential evolution algorithm (DE) is one of the most powerful stochastic real-parameter optimization algorithms. The theoretical studies on DE have gradually attracted the attention of more and more researchers. However, few theoretical researches have been done to deal with the convergence conditions for DE. In this paper, a sufficient condition and a corollary for the convergence of DE to the global optima are derived by using the infinite product. A DE algorithm framework satisfying the convergence conditions is then established. It is also proved that the two common mutation operators satisfy the algorithm framework. Numerical experiments are conducted on two parts. One aims to visualize the process that five convergent DE based on the classical DE algorithms escape from a local optimal set on two low dimensional functions. The other tests the performance of a modified DE algorithm inspired of the convergent algorithm framework on the benchmarks of the CEC2005.

1. Introduction

The differential evolution algorithm (DE) is a population-based stochastic parallel evolutionary algorithm. DE emerged as a very competitive form of evolutionary computing since it was proposed by Storn and Price in 1995 [1]. DE or its variants have been achieving competitive ranking in various competitions held on the IEEE Congress on Evolutionary Computation (CEC) Conference Series [2, 3]. According to frequently reported comprehensive studies [4–6], DE outperforms many other optimization methods in terms of convergence speed and robustness over common benchmark functions. Compared to most other evolutionary algorithms, DE is much more simple and straightforward to implement, and has very few control parameters. Perhaps due to these advantages, it has got many practical applications, such as function optimization [7–11], multiobjective optimization [12], classification [13], and scheduling [14].

Theoretical studies of algorithms are very important to understand their search behaviors and to develop more efficient algorithms. With the popularity of DE in applications, more and more researchers pay attention to the theoretical

studies on DE. According to the research contents, the main results of theoretical studies on DE can be divided into three classes as follows.

1.1. Researches on the Timing Complexity of DE. DE is a population-based stochastic search algorithm. Its runtime-complexity analysis is a critical issue. Zielinski et al. [15] investigated the runtime complexity of DE for various stopping criteria including a fixed number of generations (G_{\max}) and maximum distance criterion (*MaxDist*). *MaxDist* means that algorithms stop the execution if the maximum distance from every vector to the best population member is below a given threshold.

1.2. Researches on the Dynamical Behavior of DE's Population. This class focuses on investigating the evolving process of DE's population. For instance, the development of the expected population variance and population distribution over time is an important issue. Zaharie [16–20] theoretically analyzed the influence of the variation operators (mutation and crossover) and their parameters on the expected population variance. In 2009, Zaharie [21] theoretically investigated

the influence of the crossover operators (including classical binomial and exponential strategies) and the crossover probability on the expected population variance. Dasgupta et al. [22, 23] proposed a mathematical model of the underlying evolutionary dynamics of a one-dimensional DE-population, and the model showed that the fundamental dynamics of the each parameter vector in DE employs the gradient-descent type search strategy. Wang and Huang [24] developed a stochastic model of a one-dimensional DE-population to analyze the evolving process of the population distribution over time.

1.3. Researches on the Convergence Property of DE. This class investigates the limit behavior of DE's population. The main issue is that under which assumptions can it be guaranteed that DE or its variants can reach an optimal solution [25]. Technically speaking, commonly used concepts include convergence in probability, almost sure convergence, and convergence in distribution.

Xue et al. [26] performed a mathematical modeling and convergence analysis of continuous multi-objective differential evolution (MODE) under certain simplified assumptions, and this work was extended in [27]. Zhao et al. [28] proposed a hybrid differential evolution with transform function (HtDE) and proved its convergence. Sun [29] developed a Markov chain modeling and proved that the classical DE does not hold with convergence in probability. He et al. [30] defined the differential operator (DO) as a random mapping from the solution space to the Cartesian product of solution space and analyzed the asymptotic convergence of DE by using the random contraction mapping theorem. Ghosh et al. [31] established asymptotic convergence behavior of a classical DE (DE/rand/1/bin) algorithm by applying the concepts of Lyapunov stability theorems. And the analysis is based on the assumption that the objective function has the following two properties, (1) the objective function has the second-order continual derivative in the search space, and (2) it possesses a unique global optimum in the range of search.

The studies of this paper are confined to the third class, convergence property of DE.

We note that the conclusion of [30, 31] is in contradiction with [29]. According to the inference process, the asymptotic convergence in [30] refers to almost sure convergence. In fact, if DE does not hold with convergence in probability, then it does not hold with almost sure convergence. We also note that the value of the random mapping DO defined in [30] may be greater than 1, which is debatable. In [31], the asymptotic convergence analysis of DE/rand/1/bin, which was proved by applying Lyapunov stability theorems, should be a local convergence property. The reason is, according to Lyapunov stability theorems, the distribution of the initial population depends on the maximum region of the asymptotic stability. So for some functions, DE/rand/1/bin possesses asymptotic stability property if and only if initial individuals are closed enough to the global optimum. In addition, from the mutation operators of the classical DE, it can be derived that DE, if its population traps in a local optimum, cannot escape. This property was employed by [29] to prove that the classical DE does not possess global convergence in probability.

Taking into account that a convergent algorithm may have stronger robustness than a divergent one. Zhao et al. [28] developed a convergent algorithm, HtDE and proved its convergence. Zhan and Zhang [32] proposed a DE with random walk. Xue et al. [26, 27] analyzed MODE's convergence. However, the conditions for global convergence of DE have not been explored. In this paper, the following problems will be addressed.

- (i) What are sufficient conditions for the global convergence of DE?
- (ii) What is the algorithm framework of the convergent DE?
- (iii) Which operators can assist the classical DE to hold with a certain asymptotic convergence?

The discussion in this paper will be undertaken in a general measurable space, and *infinite production* will be used as an analysis tool.

This paper is organized as follows. Section 2 introduces the classical DE. Section 3 proves a sufficient condition and a corollary for the convergence of DE to the global optima. Section 4 presents a DE algorithm framework satisfying the convergence conditions. Section 5 proves several operators satisfying the convergent algorithm framework. Section 6 gives numerical experiments to verify the robustness of the convergent DE. Section 7 analyzes and discusses the theoretical conclusions and the experimental results in detail. Section 8 summarizes this paper and indicates several directions for future research.

2. Classical Differential Evolution

DE is a competitive algorithm for solving continuous optimization problem. Consider the optimization problem:

$$\max \{f(x); x \in S\}, \quad (1)$$

where S is a measurable space and $f(x)$ is the objective function (or the fitness of x) which satisfies that for any bounded $x \in S$, $f(x)$ is bounded. The optimal solution set is denoted as $S^* = \{x^* \mid f(x^*) = \max\{f(x)\}, x \in S\}$, where x^* is the optimum solution.

Let $\mu(\cdot)$ be a measure to space S . Perhaps $\mu(S^*) = 0$, which means that S^* is a set with measure 0. This is not convenient to analyze. In view of the accuracy of practical problems, without loss of generality, we can consider an expanded set $S_\delta^* = \{x \mid f(x^*) - f(x) < \delta\}$, where δ is a small positive value. We can choose an appropriate δ , which can meet the accuracy and make $\mu(S_\delta^*) > 0$. We use S_δ^* ($\mu(S_\delta^*) > 0$) to replace the set S^* in this paper. Meanwhile, in order to simplify the calculation, let us suppose that the search space $S = I^n$, where $I = [0, 1]$, n is the dimension of S .

The classical DE [2, 33, 34] works through a simple cycle of *reproduction* and *selection* operators after *initialization*. The reproduction operator includes *mutation* and *crossover* operators. The classical DE for solving the above problem (1) can be described in detail as follows.

- (1) Initialization: Generate an initial population denoted by $X(0)$, and let $t \leftarrow 0$.

- (2) Reproduction: Generate a trial population $Y(t)$ from the target population $X(t)$.

Mutation: generate a new population from $X(t)$ by a mutation operator, denoted by $V(t)$.

Crossover: generate a new population from $X(t)$ and $V(t)$ by a crossover operator, denoted by $U(t)$, and let $Y(t) \leftarrow U(t)$.

- (3) Selection: generate a new population from $X(t)$ and $Y(t)$ by a selection operator, denoted by $Z(t)$.
- (4) If the termination condition is satisfied, then stop; else let $t \leftarrow t + 1$ and $X(t) \leftarrow Z(t)$; then go to Step 2.

The initial population is generated by assigning random values in the search space to the variables of every solution.

2.1. Reproduction Operator

2.1.1. Mutation Operator. After initialization, DE creates a donor vector v_i^t corresponding to each individual x_i^t in the t th generation through the mutation operator. Several most frequently referred mutation strategies are presented as follows:

$$\begin{aligned} \text{DE/rand/1: } v_i^t &= x_{r_1}^t + F(x_{r_2}^t - x_{r_3}^t), \\ \text{DE/best/1: } v_i^t &= x_{\text{best}}^t + F(x_{r_1}^t - x_{r_2}^t), \\ \text{DE/cur-to-best/1: } v_i^t &= x_i^t + F(x_{\text{best}}^t - x_i^t) + F(x_{r_1}^t - x_{r_2}^t), \\ \text{DE/best/2: } v_i^t &= x_{\text{best}}^t + F(x_{r_1}^t - x_{r_2}^t) + F(x_{r_3}^t - x_{r_4}^t), \\ \text{DE/rand/2: } v_i^t &= x_{r_1}^t + F(x_{r_2}^t - x_{r_3}^t) + F(x_{r_4}^t - x_{r_5}^t), \end{aligned}$$

where x_{best}^t denotes the best individual of the current generation, the indices $r_1, r_2, r_3, r_4, r_5 \in S_r = \{1, 2, \dots, \text{NP}\} \setminus \{i\}$ are uniformly random integers mutually different and distinct from the running index i , NP is population size, and $F \in (0, 1]$ is a real parameter, called *mutation factor* or *scaling factor*.

2.1.2. Crossover Operator. Following mutation, the *crossover* operator is applied to further increase the diversity of the population. In crossover, the target vector, x_i^t , is combined with elements from the donor vector, v_i^t , to produce the trial vector, u_i^t , using the *binomial crossover*,

$$u_{i,j}^t = \begin{cases} v_{i,j}^t, & \text{if } \text{rand}(0, 1) \leq \text{Cr} \text{ or } j = j_{\text{rand}}, \\ x_{i,j}^t, & \text{otherwise,} \end{cases} \quad (2)$$

where $\text{Cr} \in (0, 1)$ is the probability of crossover and j_{rand} is a random integer in $[1, n]$. Unless otherwise mentioned, $\text{rand}(0, 1)$ is a uniformly distributed random number confined in the range $[0, 1]$.

2.2. Selection Operator. Finally, the selection operator is employed to maintain the most promising trial individuals in the next generation. The classical DE adopts a simple selection scheme. It compares the objective values of the target vector x_i^t and trial vector u_i^t . If the trial individual

reduces the value of the objective function then it is accepted for the next generation; otherwise the target individual is retained in the population. The selection operator is defined as

$$x_i^{g+1} = \begin{cases} u_i^t, & \text{if } f(u_i^t) < f(x_i^t), \\ x_i^t, & \text{otherwise.} \end{cases} \quad (3)$$

3. Convergence Condition

There are different kinds of definitions of convergence for analyzing asymptotic convergence of algorithms. The following definition of convergence, that is, convergence in probability, is used in this paper.

Definition 1. Let $\{X(t), t = 0, 1, 2, \dots\}$ be a population sequence generated by using DE to solve the optimization problem (1). Then DE converges to the global optimum, if and only if

$$\lim_{t \rightarrow \infty} p \{X(t) \cap S_\delta^* \neq \emptyset\} = 1. \quad (4)$$

Let us give a sufficient condition for the convergence of DE.

Theorem 2. Consider using DE to solve the optimization problem (1). In the t_k th target population $X(t_k)$, there exists at least one individual x , which corresponds to the trial individual y by a reproduction operator, such that

$$p \{y \in S_\delta^*\} \geq \zeta(t_k) > 0, \quad (5)$$

and the series $\sum_{k=1}^{\infty} \zeta(t_k)$ diverges; then DE converges to the optimal solution set S_δ^* .

Where $\{t_k, k = 1, 2, \dots\}$ denotes any subsequence of natural number set, $p\{y \in S_\delta^*\}$ denotes the probability that y belongs to the optimal solution set S_δ^* , and $\zeta(t_k)$ is a small positive value which may change as t_k .

Proof. In DE, each target individual corresponds to a trial individual by its reproduction operator. According to the condition of Theorem 2, we can get the probability that all the individuals of the t_k th trial population $Y(t_k)$ do not belong to the optimal solution set S_δ^* :

$$p \{Y(t_k) \cap S_\delta^* = \emptyset\} \leq 1 - \zeta(t_k), \quad (6)$$

so, we can get the probability that all the individuals of every trial population in previous $(t_k - 1)$ iterations do not belong to the optimal solution set S_δ^* :

$$\prod_{i=1}^{t_k-1} p \{Y(i) \cap S_\delta^* = \emptyset\} \leq \prod_{i=1}^{t_k-1} (1 - \zeta(i)). \quad (7)$$

And because of the elitist selection operation in DE, the optimal individual of trial populations will retain the next

generation population. So we can get the probability that the t_k th population $X(t_k)$ does not contain optima:

$$\begin{aligned} p\{X(t_k) \cap S_\delta^* = \emptyset\} &= \prod_{i=1}^{t_k-1} p\{Y(i) \cap S_\delta^* = \emptyset\} \\ &\leq \prod_{i=1}^{k-1} (1 - \zeta(i)). \end{aligned} \quad (8)$$

So for the classical DE with elitist selection, we have

$$\begin{aligned} \lim_{t \rightarrow \infty} p\{X(t) \cap S_\delta^* \neq \emptyset\} &\geq \lim_{k \rightarrow \infty} p\{X(t_k) \cap S_\delta^* \neq \emptyset\} \\ &= 1 - \lim_{k \rightarrow \infty} p\{X(t_k) \cap S_\delta^* = \emptyset\} \\ &= 1 - \lim_{k \rightarrow \infty} \prod_{i=1}^{t_k-1} p\{Y(i) \cap S_\delta^* = \emptyset\} \\ &\geq 1 - \lim_{k \rightarrow \infty} \prod_{i=1}^{k-1} (1 - \zeta(i)) \\ &= 1 - \prod_{i=1}^{+\infty} (1 - \zeta(i)). \end{aligned} \quad (9)$$

And from the property of the infinite product [35]:

$$\text{if series } \sum_{i=1}^{+\infty} \zeta(i) \text{ diverges, then } \prod_{i=1}^{+\infty} (1 - \zeta(i)) = 0. \quad (10)$$

So for the divergent series $\sum_{i=1}^{+\infty} \zeta(i)$, we can get that

$$\lim_{t \rightarrow \infty} p\{X(t) \cap S_\delta^* \neq \emptyset\} = 1. \quad (11)$$

According to Definition 1, this theorem holds. \square

Corollary 3. In Theorem 2, if $\zeta(t)$ equals ever to a positive constant $\zeta > 0$, then DE converges to the optimal solution S_δ^* .

Proof. Obviously, the series $\sum_{t=0}^{+\infty} \zeta(t)$ diverges when $\zeta(t)$ equals ever to a positive constant $\zeta > 0$. From Theorem 2, we can get that DE converges to the optimal solution S_δ^* . \square

Now we give several observations to the above conditions as follows.

- (i) Theorem 2 means that if the probability entering into the optimal set in a certain sub-sequence population is large enough, then the modified DE converges to the global optimal set in probability. And the population states need no ergodicity.
- (ii) Corollary 3 is just a special case of Theorem 2 and is very easy to check. There are some improved DE algorithms such as HtDE proposed by Zhao et al. [28], DE-RW proposed by Zhan and Zhang [32], DE-MC proposed by Braak [36], which satisfies the convergence condition of Theorem 2 (or Corollary 3).

- (iii) He and Yu [37] and Rudolph [38] presented several important conclusions on convergence conditions for evolutionary algorithms. These conclusions do apply to DE algorithm. However, comparing with these conclusions, Theorem 2 is more relaxed and easier to check.

4. Algorithm Framework Possessing Convergence

As the introduction section analyzed, it cannot be guaranteed that the classical DE holds with the global convergence. However, DE can converge to the global optimal solution if its reproduction operation satisfies the sufficient conditions given in Theorem 2 or Corollary 3. A DE algorithm framework integrating an extra mutation component will be given in this section. Owing to the fact that the purpose of using the extra mutation is to assist the classical DE to converge, this paper addresses to the operator as AsCo-mutation operator.

According to the sufficient conditions proved above, we can define the AsCo-mutation operator as follow.

Definition 4. AsCo-mutation is a mutation operator assisting the classical DE to converge. It satisfies the following conditions.

- (1) To a certain sub-sequence $\{X(t_k), k = 1, 2, \dots\}$ of population sequence $\{X(t), t = 1, 2, \dots\}$, AsCo-mutation changes at least one individual in each $\{X(t_k), k = 1, 2, \dots\}$ with a positive probability.
- (2) Let $Y(t_k)$ denote the population generated by using AsCo-mutation; there exists at least one individual y in $Y(t_k)$, such that

$$p\{y \in S_\delta^*\} \geq \zeta(t_k) > 0, \quad (12)$$

and the series $\sum_{k=1}^{+\infty} \zeta(t_k)$ diverges.

Taking into account the fact that the algorithm framework using AsCo-mutation will contain some convergent algorithms of DE family, this paper addresses to the algorithm framework as CDE. The algorithm framework CDE can be described as follows.

- (1) Initialization: generate an initial population denoted by $X(0)$, and let $t \leftarrow 0$.
- (2) Reproduction: generate a trial population $Y(t)$ from the target population $X(t)$.

Mutation: generate a new population from $X(t)$ by a mutation operator, denoted by $V(t)$.

Crossover: generate a new population from $X(t)$ and $V(t)$ by a crossover operator, denoted by $U(t)$.

AsCo-mutation: if the certain condition generating sub-sequence population is satisfied, then generate a new population $W(t)$ from $U(t)$ by AsCo-mutation and let $Y(t) \leftarrow W(t)$; otherwise, let $Y(t) \leftarrow U(t)$.

- (3) Selection: generate a new population from $X(t)$ and $Y(t)$ by a selection operator, denoted by $Z(t)$.
- (4) If the termination condition is satisfied, then stop; else let $t \leftarrow t + 1$ and $X(t) \leftarrow Z(t)$; then go to Step 2.

On the basis of DE, the reproduction operator of CDE increases a step, AsCo-mutation. Obviously, the algorithm framework CDE satisfies Theorem 2 when the AsCo-mutation satisfies the Definition 4. That is to say, CDE, which employs the AsCo-mutation given by the Definition 4, converges to the global optimum.

5. Several Mutation Operators Satisfying Convergence Condition

Like DE algorithm, most evolutionary algorithms for numerical optimization problems use vectors of floating point numbers for their chromosomal representations. For such representations, many mutation operators [39] have been proposed. The most common mutation operators include Uniform mutation [40] and Gaussian mutation [41, 42]. We introduce these operators and prove that they meet the definition of AsCo-mutation for CDE in turn.

5.1. Uniform Mutation. Uniform mutation replaces the solution vector x with a uniformly distributed random vector ξ confined in the domain S ($S = I^n$). Each component of the vector ξ is a uniformly distributed (independent identically distributed) random number from $[0, 1]$. So the density function of ξ can be expressed as:

$$p_{\xi}(x) = \begin{cases} 1, & \text{if } x \in I^n, \\ 0, & \text{otherwise.} \end{cases} \quad (13)$$

As shown in the CDE algorithm framework, suppose that AsCo-mutation operator employed by CDE is Uniform mutation. Let w denote the new individual generated by Uniform mutation; then the probability that w belongs to the optimal solution set can be calculated as follow:

$$p\{w \in S_{\delta}^*\} = \int_{S_{\delta}^*} p_{\xi}(x) dm = \mu(S_{\delta}^*) > 0. \quad (14)$$

The method that CDE uses Uniform mutation is flexible, such as mutating an arbitrary individual selected from the set $U(t)$ at a given probability P_{ac} and mutating more than one individual. Let k ($k < NP$) denote the number of mutated individuals, then the probability $P_{one/W}$ that at least one of $W(t)$ belongs to the optimal solution set can be calculated as follow:

$$\begin{aligned} P_{one/W} &\geq 1 - [1 - p\{w \in S_{\delta}^*\} \cdot P_{ac}]^k \\ &= 1 - [1 - \mu(S_{\delta}^*) \cdot P_{ac}]^k > 0, \end{aligned} \quad (15)$$

where the P_{ac} is an empirical probability, $P_{ac} \in (0, 1]$, and the diversity of the population will gradually enhance as P_{ac} increases.

In addition, the implementation of Uniform mutation operator can be also flexible. For example, in order to keep the tradeoff between exploration and exploitation, this paper presents the following operator.

DE/um-best/1

$$v_i^t = x_i^t + F(x_{best}^t - x_i^t) + \text{rand}(0, 1)(x_{r_{b_1}}^t - x_{r_{b_2}}^t), \quad (16)$$

where $\text{rand}(0,1)$ denotes a uniform random number in $[0, 1]$. The $x_{r_{b_1}}, x_{r_{b_2}}$ are boundary individuals at a given probability P_b , each element of which equals either the upper boundary or the lower boundary value. The r_{b_1}, r_{b_2} are uniform random integers in $[1, \lfloor NP(1 + P_b) \rfloor]$. That is, when the index $r_{b_1} (r_{b_2})$ is no less than NP , $x_{r_{b_1}} (x_{r_{b_2}})$ will takes a boundary individual. Obviously, if $x_{r_{b_1}, j}$ takes the upper boundary value of the j th dimension while $x_{r_{b_2}, j}$ takes the lower boundary value (and vice-versa), then the element $v_{i,j}^t$ is ergodic in the j th dimension. Therefore the individual v_i^t can be ergodic in the search space like Uniform mutation operator.

5.2. Gaussian Mutation. Gaussian mutation modifies all components of the solution vector x by adding a random noise:

$$\tilde{x} = x + \eta, \quad (17)$$

where η is a vector of independent random Gaussian numbers with a mean of zero and standard deviations σ . The density function of η can be expressed as:

$$\begin{aligned} p_{\eta}(y) &= \left(\frac{1}{\sigma\sqrt{2\pi}}\right)^n \cdot \prod_{i=1}^n \exp\left(\frac{-y_i^2}{2\sigma^2}\right) \\ &= \left(\frac{1}{\sigma\sqrt{2\pi}}\right)^n \cdot \exp\left(-\sum_{i=1}^n \frac{y_i^2}{2\sigma^2}\right). \end{aligned} \quad (18)$$

Now, let us suppose that AsCo-mutation operator employed by CDE is Gaussian mutation. Then the probability that w generated by Gaussian mutation belongs to the optimal solution set can be calculated as follow:

$$p\{w \in S_{\delta}^*\} = \int_{\tilde{S}_{\delta}^*} p_{\eta}(y) dm, \quad (19)$$

where $\tilde{S}_{\delta}^* = S_{\delta}^* - x = \{z - x \mid z \in S_{\delta}^*\}$.

On the other hand, for any individual $x \in S = I^n$, such that $\tilde{S}_{\delta}^* \subseteq [-1, 1]^n$. So

$$p_{\eta}(y) \geq \left(\frac{1}{\sigma\sqrt{2\pi}}\right)^n \cdot \exp\left(-\frac{n}{2\sigma^2}\right), \quad \forall y \in \tilde{S}_{\delta}^*. \quad (20)$$

Implying that

$$\begin{aligned} p\{w \in S_{\delta}^*\} &\geq \mu(\tilde{S}_{\delta}^*) \cdot \left(\frac{1}{\sigma\sqrt{2\pi}}\right)^n \cdot \exp\left(-\frac{n}{2\sigma^2}\right) \\ &= \mu(S_{\delta}^*) \cdot \left(\frac{1}{\sigma\sqrt{2\pi}}\right)^n \cdot \exp\left(-\frac{n}{2\sigma^2}\right) \\ &> 0. \end{aligned} \quad (21)$$

Like uniform mutation, the used method of Gaussian mutation is flexible. As before k ($k < NP$) denotes the number of individuals mutated by Gaussian mutation operator, P_{ac} denotes the probability that each individual is mutated, and $P_{one/W}$ denotes the probability that at least one of $W(t)$ belongs to the optimal solution set. Then $P_{one/W}$ can be calculated as follow:

$$\begin{aligned} P_{one/W} &\geq 1 - [1 - p\{w \in S_{\delta}^*\} \cdot P_{ac}]^k \\ &\geq 1 - \left[1 - \mu(S_{\delta}^*) \cdot \left(\frac{1}{\sigma\sqrt{2\pi}}\right)^n \cdot \exp\left(-\frac{n}{2\sigma^2}\right)\right]^k \\ &> 0. \end{aligned} \quad (22)$$

Obviously, let $\zeta(t) = P_{one/W}$, uniform mutation and Gaussian mutation operators satisfy Definition 4. And thus we can get the following Theorem 5.

Theorem 5. *DE algorithm employing uniform mutation or Gaussian mutation operator converges in probability to the global optimum of the optimization problem (1).*

6. Experimental Verification

It is proved in the previous sections that CDE algorithms possess convergence in probability, which only means it can be guaranteed that CDE algorithms reach an optimal solution when the iteration times approaches infinity, but does not mean that CDE can find out the optimal solution within finite iteration times. However, a convergent algorithm should generally hold stronger robustness. Thus this section gives experiments by being composed of two parts to verify CDE's robustness. One aims to visualize the process escaping from a local optimal set of CDE on two low dimensional functions. The other is conducted to test a modified DE algorithm, which is inspired of the above convergence theory, on the benchmark functions of the CEC2005.

6.1. Experiments on Low Dimensional Functions. To achieve the aims mentioned above, experiments are conducted on two numerical functions which are chosen according to the experimental results of [43–46]. One is the DE deceptive function [45], which can lead the classical DE to trap in the local optimum. The other is the Rastrigin function. In [45, 46], nineteen benchmark functions including the rastrigin function were tested using the classical DE. Those results indicated that the optimization effect of the rastrigin function is one of the worst.

6.1.1. Deceptive Function. Consider

$$f(x) = \begin{cases} -3\text{sinc}(2x + 10), & \text{if } -10 \leq x < 0, \\ -\sqrt{x} \cdot \sin(x\pi), & \text{if } 0 \leq x \leq 10, \end{cases} \quad (23)$$

where the function $\text{sinc}(t)$ is given by

$$\text{sinc}(t) = \begin{cases} 1, & \text{if } t = 0, \\ \frac{\sin \pi t}{\pi t}, & \text{if } t \neq 0. \end{cases} \quad (24)$$

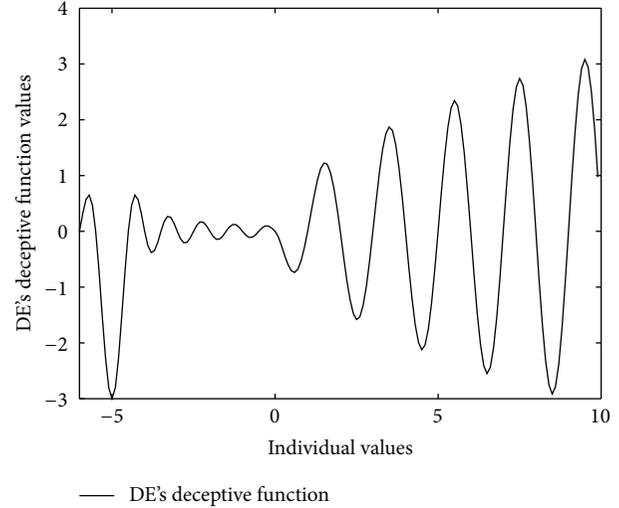


FIGURE 1: DE's deceptive function.

The landscape of DE deceptive function is shown in Figure 1. The global optimum of the function is $x = -5.0$ with the function value $f(x) = -3$. There is a deceptive local minimum $x = 8.5060$ with function value $f(x) = -2.9160$ in this test function.

6.1.2. Rastrigin Function (2 Dimensions). Consider

$$\begin{aligned} f(x) &= 20 + \sum_{i=1}^2 \left((x_i)^2 - 10 \cos(2\pi x_i) \right), \\ x &\in [-5.12, 5.12]^2. \end{aligned} \quad (25)$$

The global optimum of the function is $x = (0, 0)$ with the function value $f(x) = 0$. There are many local optima in this test function.

Let CDE-um denote the CDE algorithm using uniform mutation operator. Suppose that CDE-um mutates the worst individual of $U(t)$ at probability 1, and the new individual is directly retained to the next generation. Experiments were conducted to compare five typical versions of the classic DE with CDE-um algorithm. All experiments were implemented for 50 independent replications. The convergence times and convergence ratio on the 50 replications were reported.

In order to show the robustness of CDE-um, we reported the number of function evaluations (FES) to achieve the Ter_Err within Max_FES. Table 1 gave the FES of 50 independent replications of five typical versions on the DE's deceptive function, while Table 2 reported the FES on the Rastrigin function. Those typical versions included DE/best/1 versus CDE-um/best/1, DE/rand/1 versus CDE-um/rand/1, DE/cur-to-best/1 versus CDE-um/cur-to-best/1, and DE/best/2 versus CDE-um/best/2, as well as DE/rand/2 versus CDE-um/best/2. Table 3 analyzes the results of Tables 1 and 2. From the statistics of Table 3, we can see that the ratio (ConRa) converging to the optimum of CDE-um is much higher than the corresponding DE.

TABLE 1: FES to achieve the Ter_Err for DE's deceptive function.

Alg.	best/1		rand/1		cur-to-best/1		best/2		rand/2	
	DE	CDE-um	DE	CDE-um	DE	CDE-um	DE	CDE-um	DE	CDE-um
FES	176	192	272	416	448	824	264	384	392	632
	—	4240	—	184032	—	3328	—	424	560	504
	—	147504	—	3920	—	2160	272	744	560	504
	—	500832	304	464	—	1384	—	1808	336	1296
	—	26200	—	312	—	560	—	1552	536	600
	—	163976	448	6248	—	1360	—	320	—	808
	—	1568	368	592	—	464	296	312	472	888
	—	200	368	727168	—	2944	—	464	560	936
	—	5448	320	312	592	664	392	912	424	472
	—	66712	—	312	—	22736	304	392	—	768
	—	98824	320	5040	—	520	272	1448	352	744
	—	82080	—	177312	344	240	—	416	504	648
	—	50208	288	520	592	4272	—	368	—	800
	—	5408	272	320	—	824	272	408	360	704
	—	3096	352	400	416	976	288	344	576	680
	—	152256	304	352	—	832	288	320	—	528
	—	1.41E + 06	—	312	—	4200	—	3592	432	624
	—	312624	256	312	—	1192	288	256	480	672
	—	370328	224	508880	—	1232	—	608	—	672
	—	176	256	2144	—	1328	272	2600	368	784
	—	205016	—	632248	—	1240	312	352	376	592
	184	168	328	360	672	376	328	464	392	720
	—	607768	312	688	—	2888	—	416	408	672
	—	2200	—	2240	—	632	—	464	352	896
	—	184240	—	1312	536	752	312	520	472	1080
	160	406128	—	448	552	408	368	360	—	672
	—	268560	—	817456	—	9792	—	2456	448	560
	—	295208	—	360	—	1128	—	776	352	744
	—	38936	336	336	—	672	—	416	—	728
	—	160	296	402096	472	512	208	328	496	784
	—	126144	—	368	—	520	336	4360	—	576
	—	759336	312	184160	752	2456	—	376	—	840
	—	292232	304	2064	392	680	—	480	320	5032
	—	24032	—	336	632	936	272	2088	352	656
	—	240	—	107664	—	2456	272	2696	456	552
	—	555152	—	1432	—	1008	320	1096	336	1664
	168	265536	264	105600	—	352	232	512	—	752
	168	2256	296	1864	—	1680	360	352	384	760
	—	55616	280	320	—	20664	—	904	—	832
	—	1.11E + 06	336	248	—	592	—	376	352	824
	176	263744	—	11136	—	800	408	448	616	69016
	—	123896	352	400	456	2168	264	960	536	592
	—	200	352	1456	288	672	288	384	544	792
	—	784008	304	238752	—	1368	320	5320	520	2536
	—	164408	—	29432	—	1072	304	1248	488	584
208	252816	—	4216	392	704	—	568	496	536	
—	2088	—	3200	432	560	280	440	576	536	
—	542560	—	312	616	472	440	368	—	848	
192	1768	—	80536	—	640	—	432	—	672	
—	743632	304	320	—	592	—	2976	408	680	

“FES” denotes the number of function evaluations. “—” indicates that the algorithm cannot find the optimum within Max_FES.

TABLE 2: FES to achieve the Ter_Err for Rastrigin function.

Alg.	best/1		rand/1		cur-to-best/1		best/2		rand/2	
	DE	CDE-um	DE	CDE-um	DE	CDE-um	DE	CDE-um	DE	CDE-um
—	—	103376	—	1680	1184	824	1088	1232	2000	2400
—	—	1.60E + 06	1392	1472	1408	3328	976	1392	2112	2432
560	—	1.19E + 06	1664	1424	1264	2160	1264	1392	1584	2512
—	—	942224	—	1200	1392	1384	1168	1280	1936	2176
—	—	672	1440	1584	1280	560	1136	1280	1792	2480
—	—	452064	1696	1600	1184	1360	1120	1312	1856	2304
560	—	624	1536	1456	1600	464	1088	1184	2000	1792
720	—	688	1424	1648	1280	2944	—	1472	—	2352
—	—	576	1296	1552	1312	664	—	1296	1920	2128
672	—	576	1184	1248	1184	22736	1056	1296	1776	1936
—	—	198368	1424	1424	2720	520	1072	1280	2176	2368
—	—	60336	—	1856	1104	240	1200	1280	2000	2480
656	—	5.27E + 06	1424	1584	1136	4272	896	5840	1840	2304
—	—	267360	1552	28192	1376	824	1056	1056	1728	2288
—	—	3.16E + 06	1568	1344	1376	976	1136	1408	1792	2720
672	—	1.11E + 06	1296	1440	—	832	1136	1552	2416	2144
608	—	800	1408	1856	1200	4200	800016	1216	1936	2240
672	—	2.54E + 06	1600	1600	1456	1192	1088	1568	2112	2496
704	—	3.75E + 06	1536	3712	1184	1232	1168	1232	2096	2512
560	—	236592	1328	1648	1344	1328	1072	1408	1904	2240
—	—	3.91E + 06	1536	1808	1952	1240	—	1328	2336	2368
—	—	241776	1200	1872	1280	376	1216	1312	1744	2880
—	—	1072	1408	218208	1344	2888	1152	1248	2000	2512
—	—	4.70E + 06	1520	1408	1232	632	1264	1152	1904	2464
FES	—	8.20E + 06	1344	689360	1136	752	1200	5568	1856	2352
—	—	640	1424	1488	1376	408	1184	1424	1808	2448
—	—	788880	1344	3.49E + 06	1376	9792	1216	1424	2416	2512
576	—	656	1408	1168	1056	1128	1168	1296	1856	2240
640	—	560	1408	1536	1184	672	960	1344	1856	2432
—	—	2.74E + 06	1376	228048	5136	512	1312	1312	1888	2512
—	—	672	1312	1520	1456	520	1232	1184	2112	2160
736	—	17952	—	1584	1536	2456	1200	3504	1920	2560
592	—	1.87E + 06	1264	1536	1408	680	1296	1136	2000	2432
—	—	15456	1328	24288	1280	936	1296	1264	1856	2272
624	—	2.39E + 06	1232	1680	1264	2456	1136	1152	2144	2448
—	—	1.72E + 06	1264	1504	1040	1008	1264	1504	1776	2352
544	—	656	1440	1616	1184	352	944	1200	1824	2432
640	—	5.25E + 06	1296	1520	1184	1680	960	1360	2064	2048
720	—	656	1600	2.27E + 06	1520	20664	976	1296	1744	2080
672	—	4.95E + 06	1536	1392	1120	592	1056	1232	1904	2464
—	—	784	1472	1456	1344	800	1072	1168	1776	2176
528	—	1.18E + 06	1472	9.47E + 06	1472	2168	1280	12944	1824	2672
640	—	576	1344	1648	1376	672	1056	10960	1984	2528
624	—	9.54E + 06	1680	1360	1328	1368	1120	1248	1952	2768
720	—	87056	1568	1392	1376	1072	1104	1136	1776	2464
640	—	640	1280	1696	1344	704	1088	1264	2096	2368
576	—	1.86E + 06	1328	1344	1232	560	1232	1312	2000	2240
—	—	848	—	1504	1344	472	1200	1136	2080	2176
544	—	688	1600	1440	1344	640	1056	1216	2000	2544
640	—	9.40E + 06	1392	1408	1136	592	1056	1344	2096	2336

“FES” denotes the number of function evaluations. “—” indicates that the algorithm cannot find the optimum within Max_FES.

TABLE 3: Statistical analysis of Tables 1 and 2.

Function Algorithm	Deceptive function			Rastrigin function		
	RTs	CTs	ConRa	RTs	CTs	ConRa
DE/best/1	50	8	16%	50	27	54%
CDE-um/best/1	50	50	100%	50	50	100%
DE/rand/1	50	28	56%	50	45	90%
CDE-um/rand/1	50	50	100%	50	50	100%
DE/cur-to-best/1	50	17	34%	50	49	98%
CDE-um/cur-to-best/1	50	50	100%	50	50	100%
DE/best/2	50	29	58%	50	47	94%
CDE-um/best/2	50	50	100%	50	50	100%
DE/rand/2	50	37	74%	50	49	98%
CDE-um/rand/2	50	50	100%	50	50	100%

“RTs” denotes runing times. “CTs” denotes convergence times. “ConRa” denotes convergence ratio.

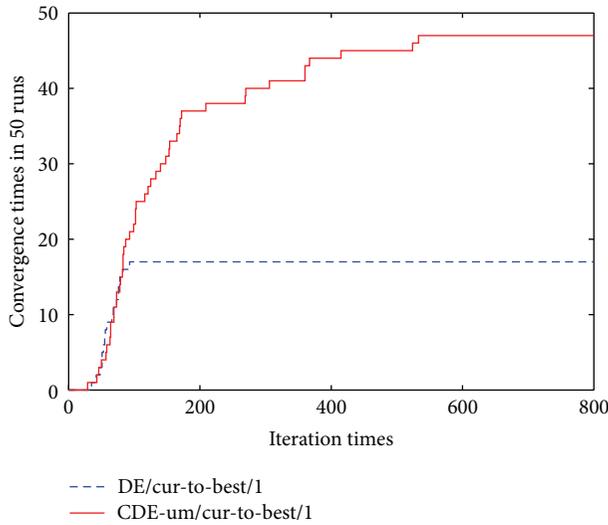


FIGURE 2: Convergence times figure in 50 runs for DE's deceptive function using */cur-to-best/1 algorithms.

Figures 2, 3, and 4 showed the convergence times graphs in 50 independent replications within 800 iterations for DE's deceptive function. We can see that all the convergence curves hold two common characteristics as follows.

- (i) When the iteration times are smaller, the convergence times of five typical versions of the classical DE are slightly larger than the corresponding CDE-um. However, with the iteration times are increasing, the convergence times of CDE-um will become far larger than the corresponding DE. From this we can see that smaller increasing in the computational cost can make a greatly improving on the robustness of CDE-um algorithm.
- (ii) When the iteration times are larger, all the convergence graphs of five typical versions of the classical DE become a straight line. However, all the graphs of CDE-um show the ladder's rising status. This indirectly shows that the classical DE cannot escape

from a local optimal set or a premature solution set if trapping in, but CDE-um enhances the ability to escape from the local optimal set or premature solution set.

The convergence graphs on the rastrigin function had the similar characteristics with DE's deceptive function, so the graphics are omitted here.

The population size is set to $8 \times n$. The maximum number of function evaluations (Max_FES) is set to 5,000,000.

- (i) Mutation factor, $F = 0.5$ [44, 47];
- (ii) Crossover probability, $Cr = 0.9$ [44, 47];
- (iii) Termination error value (Ter_Err), $Ter_Err = 10^{-12}$.

6.2. *Experiments on Functions of CEC2005.* Wang et al. [48] presented a composite differential evolution algorithm (CoDE), which employed three trial vector generation strategies, that is, rand/1/bin, rand/2/bin, and current-to-rand/1. The experimental studies on the 25 benchmark functions of CEC2005 have indicated that CoDE's overall performance was better than the other seven outstanding competitors (please refer to [48] for details). Now we give a convergent CoDE algorithm (CCoDE-umbest) based on the above convergent algorithm framework. The CCoDE-umbest algorithm has the DE/um-best/1 operator, which was presented in Section 5.1, instead of the current-to-rand/1 of CoDE.

This paper compared CCoDE-umbest with CoDE on the 25 benchmark functions of CEC2005. Table 4 reported the average and standard deviation of the function error values obtained in 25 runs when $FES = 1.5E + 5$ and $FES = 3.0E + 5$, respectively. The two bottom lines in Table 4 gave the test statistics for sign test [49] on the mean errors. From Table 4, the probability values (0.012 for $FES = 1.5E + 5$, 0.041 for $FES = 3.0E + 5$) supporting the null hypothesis are less than the significance level at 0.05. So we can reject the null hypothesis, that is to say, the overall performance of CCoDE-umbest is better than CoDE on the benchmarks. It implies that the use of the convergent algorithm framework can improve the performance of CoDE.

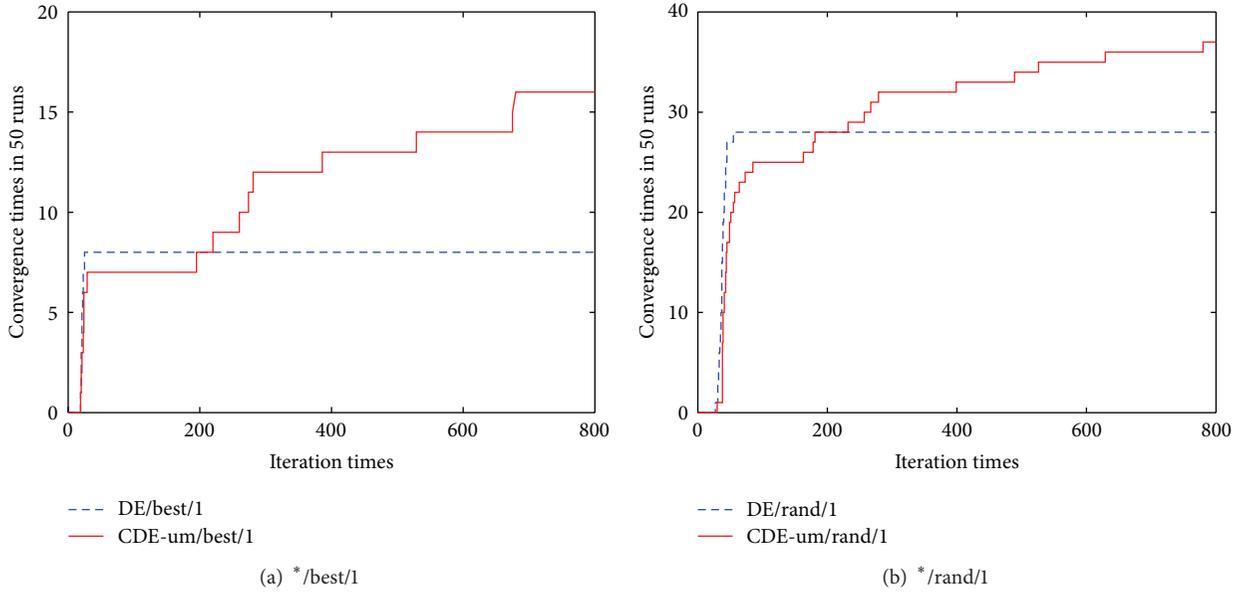


FIGURE 3: Convergence times figure in 50 runs for DE's deceptive function.

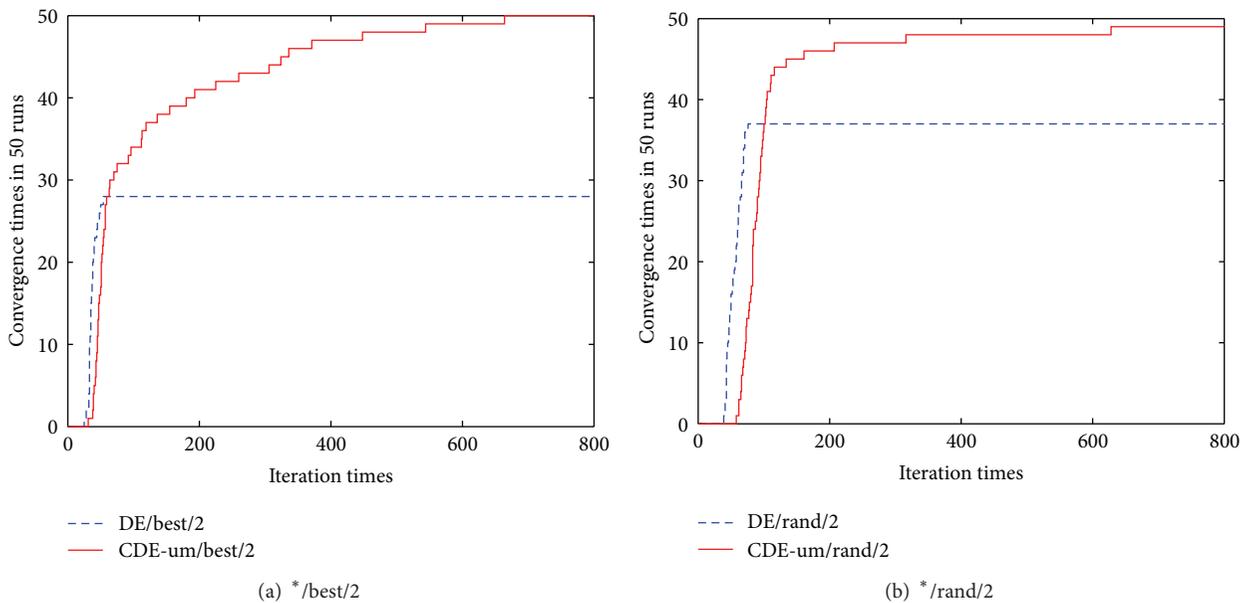


FIGURE 4: Convergence times figure in 50 runs for DE's deceptive function.

The population size was set to 60, and the dimension was set to 10. The strategies of the other parameters are the same to [48].

All the above algorithms are coded in Visual C++ and the experiments were executed on a ACER 4750G laptop with a 2.30 GHz Intel(R) Core (TM)i5 2410M CPU and 2 GB RAM.

7. Analysis and Discussion

In this paper, two sufficient conditions for the convergence of DE have been presented in forms of a theorem and a corollary. These conditions describe the limiting behaviors of DE. Given a sub-sequence population, the sufficient conditions

require that the probability generating an optimum (or optima) by the reproduction operations is greater than a small positive number. Taking into account the selection operator of DE which can retain the elitist individual(s) of current population to the next generation, the sufficient conditions were easily proved by using the infinite product. Judging by essentials, sufficient conditions for the convergence of the classical evolutionary algorithms [38] and the elitist genetic algorithm [50–52], which is proved by using the Markov Chain, generally conclude two requirements. One is the ergodicity of the population states; the other is the retention of the current best solution. In contrast, the sufficient conditions for the convergence of DE presented in this paper

TABLE 4: The function error values achieved via CCoDE-umbest and CoDE on CEC2005's test functions.

FES	1.5E + 5				3.0E + 5			
	CCoDE-umbest		CoDE		CCoDE-umbest		CoDE	
	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.
f1	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00
f2	1.83E - 04	1.21E - 04	5.16E - 04	2.28E - 04	1.14E - 11	1.13E - 11	7.42E - 11	8.58E - 11
f3	1.01E + 04	9.71E + 03	2.78E + 05	1.07E + 05	5.33E + 03	9.45E + 03	5.05E + 04	1.34E + 04
f4	8.19E - 03	5.36E - 03	1.84E - 02	1.07E - 02	1.67E - 08	1.39E - 08	7.72E - 08	6.33E - 08
f5	2.98E - 03	1.37E - 03	8.78E - 02	3.06E - 02	9.56E - 10	5.66E - 10	1.11E - 06	7.15E - 07
f6	2.88E - 05	2.40E - 05	5.62E - 03	3.56E - 03	3.41E - 14	3.22E - 14	5.91E - 09	4.29E - 09
f7	2.12E - 01	1.46E - 01	2.69E - 01	5.34E - 02	2.12E - 01	1.46E - 01	1.69E - 01	3.94E - 02
f8	2.01E + 01	7.89E - 02	2.03E + 01	5.47E - 02	2.01E + 01	7.08E - 02	2.03E + 01	6.77E - 02
f9	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00
f10	8.81E + 00	3.05E + 00	2.05E + 01	3.25E + 00	8.80E + 00	3.05E + 00	1.40E + 01	2.52E + 00
f11	1.59E + 00	9.51E - 01	6.17E + 00	6.60E - 01	1.16E + 00	1.01E + 00	5.28E + 00	8.16E - 01
f12	2.76E + 00	4.89E + 00	5.00E + 02	2.03E + 02	1.64E + 00	4.78E + 00	1.54E + 02	1.14E + 02
f13	3.40E - 01	1.63E - 01	7.11E - 01	8.70E - 02	3.12E - 01	1.48E - 01	5.01E - 01	6.40E - 02
f14	3.03E + 00	3.50E - 01	3.35E + 00	1.90E - 01	3.03E + 00	3.50E - 01	3.26E + 00	1.73E - 01
f15	0.00E + 00	0.00E + 00	3.24E + 00	1.10E + 01	0.00E + 00	0.00E + 00	3.24E + 00	1.10E + 01
f16	1.10E + 02	7.15E + 00	1.34E + 02	5.87E + 00	1.10E + 02	7.20E + 00	1.16E + 02	5.68E + 00
f17	1.10E + 02	9.83E + 00	1.55E + 02	1.02E + 01	1.09E + 02	9.71E + 00	1.38E + 02	7.95E + 00
f18	4.80E + 02	2.40E + 02	5.00E + 02	2.45E + 02	4.80E + 02	2.40E + 02	5.00E + 02	2.45E + 02
f19	5.80E + 02	2.48E + 02	4.20E + 02	2.14E + 02	5.80E + 02	2.48E + 02	4.20E + 02	2.14E + 02
f20	5.40E + 02	2.50E + 02	4.00E + 02	2.00E + 02	5.40E + 02	2.50E + 02	4.00E + 02	2.00E + 02
f21	5.16E + 02	7.84E + 01	5.00E + 02	3.41E - 13	5.16E + 02	7.84E + 01	5.00E + 02	3.41E - 13
f22	6.80E + 02	1.66E + 02	6.20E + 02	2.19E + 02	6.79E + 02	1.66E + 02	6.16E + 02	2.17E + 02
f23	5.59E + 02	1.46E - 12	5.59E + 02	1.32E - 12	5.59E + 02	1.46E - 12	5.59E + 02	3.41E - 13
f24	2.00E + 02	0.00E + 00	2.00E + 02	0.00E + 00	2.00E + 02	0.00E + 00	2.00E + 02	0.00E + 00
f25	2.00E + 02	0.00E + 00	2.00E + 02	0.00E + 00	2.00E + 02	0.00E + 00	2.00E + 02	0.00E + 00
Sign test on mean	Neg_Dif	Pos_Dif	Ties	P value	Neg_Dif	Pos_Dif	Ties	P value
	16	4	5	0.012	15	5	5	0.041

“FES” denotes the number of function evaluations. “Std.” denotes the standard deviation of 25 mean errors. The two bottom lines record the test statistics for sign test on mean errors. “Neg_Dif” and “Pos_Dif” denote the number of negative and positive differences, respectively. “P value” denotes the probability value supporting the null hypothesis. Here the both P value are less than the significance level at 0.05.

do not require the population state to hold with the ergodic property.

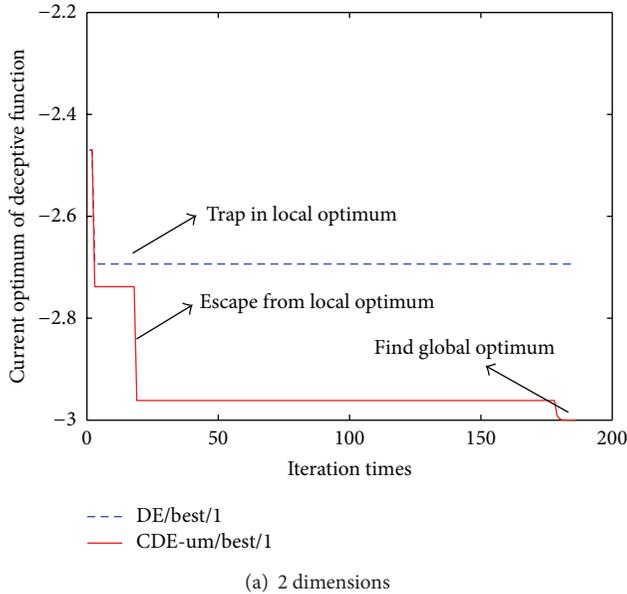
Theoretical studies of algorithms' convergence are of significance to understand their search behaviors and to develop more robust algorithms. According to the presented sufficient conditions, a modified algorithm framework, CDE, is proposed. By employing an extra mutation operator, the CDE algorithm framework becomes to converge in probability. It is not difficult to infer that there are many mutation operators meeting the convergent condition, such as uniform mutation, Gaussian mutation operator, and other mutation operators.

Thus, now there arises a new problem: which mutation operator is the most suitable one? Inspired by the process from the classical genetic algorithm to the elitist genetic algorithm, our preference should be directed to the operators with the following characteristics. Firstly, the auxiliary operator is simple and straightforward to implement. Secondly, the operator can make DE algorithm convergent, thereby improving the robustness of the algorithm. Finally, the

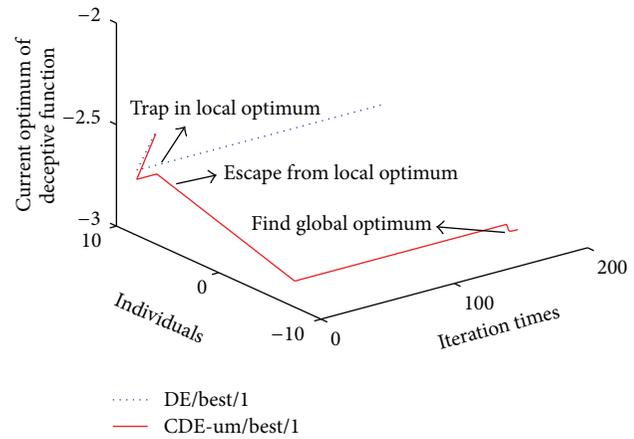
computational cost generated by the auxiliary operator is reasonable. Based on these factors, this paper presents the CDE-um and CCoDE-umbest algorithms and gives numerical experiments to verify the robustness and competitiveness of those convergent DE algorithms.

From Table 3, we can see that all the convergence ratios of five versions of CDE-um on test functions reach 100%. This shows that the convergent algorithm CDE-um improves the robustness of the classical DE. In addition, from Figures 2–4, comparing with the corresponding DE, the computational cost of CDE-um is not large, and still acceptable and reasonable. As shown in Table 4, the results on the CCoDE-umbest algorithm indicate that the reasonable use of the convergent algorithm framework can improve the performance of CoDE.

Moreover, the robustness of CDE-um also can be further analyzed by numerical experiments. Figure 5 gives the convergence graphs for DE's deceptive function in a single run. From the graphs in Figure 5, we can see that CDE-um can escape from the local optimum of the test functions, while the classical DE cannot escape in case of trapping in the local



(a) 2 dimensions



(b) 3 dimensions

FIGURE 5: Convergence graph for DE's deceptive function at a run.

optimum solution set. In fact, the mutation perturbations approach 0 when the classical DE traps in a local optimum solution set, which results in that the population of the classical DE cannot be optimized any more.

Of course, we have to note that convergence in probability is a property when the iteration times approach infinity. The previous theorem and experimental results cannot infer that the CDE can solve all function optimization problems within a finite iteration.

8. Conclusion and Future Work

The convergent property studies, as one of the basic researches for algorithms, benefit designing more robust algorithms. Few of researches have been done in dealing with conditions for the convergence of DE. This paper presented and proved two sufficient conditions for the convergence of DE. These sufficient conditions state that DE variants can guarantee converging to a global optimum solution if the probability, generating an optimum (or optima) by the reproduction operations of each generation in a certain sub-sequence population, is greater than a small positive number. According to the sufficient conditions, a convergent algorithm framework CDE was presented. The algorithm framework demonstrates that the employment of some auxiliary operators satisfying certain conditions can make the classical DE converge in probability. It was then proved that uniform mutation and Gaussian mutation operators meet the convergence conditions of the auxiliary operator.

Convergent algorithms may not always work competitively, but these should generally possess more powerful robustness. So in order to further verify the conclusions drawn from the theoretic researches, this paper gave numerical experiments comparing the performance of the convergent algorithm, CDE-um algorithm, and the classical

DE (including five typical versions). CDE-um algorithm was designed by incorporating uniform mutation into the classical DE. The experimental results on the test functions show that smaller increase in the computational cost can make a greatly improvement on the robustness for all five typical versions of CDE-um. In addition, this paper improved the *Composite Differential Evolution* (CoDE) inspired of the convergence theory and tested its competitiveness on the benchmark functions of CEC2005.

In summary, the sufficient conditions guaranteeing global convergence of DE variants, which were proved in this paper, are easy to check and are general enough to be useful for the family of DE algorithms. And in future works, it appears to be promising for developing more competitive and convergent algorithms by incorporating a certain convergence-assisted operator into some outstanding variants of modified DE algorithms.

Acknowledgments

The authors would like to acknowledge the editor and reviewers for helpful comments. This work was supported in part by the National Natural Science Foundation of China under Grant no. 61170202, the Fundamental Research Funds for the Central Universities under Grant no. 2012-YB-19, the Key Project of Chinese Ministry of Education under Grant no. 212109, and the Key Project of the Education Department of Hubei Province under Grant no. D20132701.

References

- [1] R. Storn and K. V. Price, "Differential evolution: a simple and efficient adaptive scheme for global optimization over continuous spaces," Tech. Rep., International Computer Science Institute, 1995.

- [2] S. Das and P. N. Suganthan, "Differential evolution: a survey of the state-of-the-art," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, pp. 4–31, 2011.
- [3] S. Das, P. N. Suganthan, and C. A. C. Coello, "Guest editorial special issue on differential evolution," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, pp. 1–3, 2011.
- [4] J. Vesterström and R. Thomsen, "A comparative study of differential evolution, particle swarm optimization, and evolutionary algorithms on numerical benchmark problems," in *Proceedings of the Congress on Evolutionary Computation (CEC '04)*, pp. 1980–1987, June 2004.
- [5] K. V. Price, R. M. Storn, and J. A. Lampinen, *Differential Evolution: A Practical Approach to Global Optimization*, Natural Computing Series, Springer, Berlin, Germany, 2005.
- [6] O. Hrstka and A. Kučerová, "Improvements of real coded genetic algorithms based on differential operators preventing premature convergence," *Advances in Engineering Software*, vol. 35, no. 3-4, pp. 237–246, 2004.
- [7] G. Iacca, F. Caraffini, and F. Neri, "Compact differential evolution light: high performance despite limited memory requirement and modest computational overhead," *Journal of Computer Science and Technology*, vol. 27, no. 5, pp. 1056–1076, 2012.
- [8] R. J. He and Z. Y. Yang, "Differential evolution with adaptive mutation and parameter control using Lévy probability distribution," *Journal of Computer Science and Technology*, vol. 27, no. 5, pp. 1035–1055, 2012.
- [9] A. W. Mohamed, H. Z. Sabry, and M. Khorshid, "An alternative differential evolution algorithm for global optimization," *Journal of Advanced Research*, vol. 3, no. 2, pp. 149–165, 2012.
- [10] A. W. Mohamed and H. Z. Sabry, "Constrained optimization based on modified differential evolution algorithm," *Information Sciences*, vol. 194, pp. 171–208, 2012.
- [11] E. Mezura-Montes, M. E. Miranda-Varela, and R. del Carmen Gómez-Ramón, "Differential evolution in constrained numerical optimization: an empirical study," *Information Sciences*, vol. 180, no. 22, pp. 4223–4262, 2010.
- [12] A. M. Gujarathi and B. V. Babu, "Hybrid multi-objective differential evolution (H-MODE) for optimization of polyethylene terephthalate (PET) reactor," *International Journal of Bio-Inspired Computation*, vol. 2, no. 3-4, pp. 213–221, 2010.
- [13] X. Lu and K. Tang, "Classification- and regression-assisted differential evolution for computationally expensive problems," *Journal of Computer Science and Technology*, vol. 27, no. 5, pp. 1024–1034, 2012.
- [14] Q. K. Pan, L. Wang, and B. Qian, "A novel differential evolution algorithm for bi-criteria no-wait flow shop scheduling problems," *Computers and Operations Research*, vol. 36, no. 8, pp. 2498–2511, 2009.
- [15] K. Zielinski, D. Peters, and R. Laur, "Run time analysis regarding stopping criteria for differential evolution and particle swarm optimization," in *Proceedings of the 1st International Conference on Experiments Process System Modelling, Simulation and Optimization*, 2005.
- [16] D. Zaharie, "On the explorative power of differential evolution," in *Proceedings of the 3rd International Workshop on Symbolic Numerical Algorithms for Scientific Computing*, October 2001.
- [17] D. Zaharie, "Critical values for the control parameters of differential evolution algorithms," in *Proceedings of the 8th International Mendel Conference on Soft Computing*, pp. 62–67, 2002.
- [18] D. Zaharie, "Parameter adaptation in differential evolution by controlling the population diversity," in *Proceedings of the 4th International Workshop on Symbolic Numeric Algorithms for Scientific Computing*, pp. 385–397, 2002.
- [19] D. Zaharie, "A comparative analysis of crossover variants in differential evolution," in *Proceedings of the International Multiconference on Computer Science and Information Technology (IMCSIT '07)*, pp. 171–181, Wisla, Poland, 2007.
- [20] D. Zaharie, "Statistical properties of differential evolution and related random search algorithms," in *Proceedings of the International Conference on Computational Statistics (COMPSTAT '08)*, pp. 473–485, 2008.
- [21] D. Zaharie, "Influence of crossover on the behavior of differential evolution algorithms," *Applied Soft Computing Journal*, vol. 9, no. 3, pp. 1126–1138, 2009.
- [22] S. Dasgupta, S. Das, A. Biswas, and A. Abraham, "The population dynamics of differential evolution: a mathematical model," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '08)*, pp. 1439–1446, Hong Kong, China, June 2008.
- [23] S. Dasgupta, S. Das, A. Biswas, and A. Abraham, "On stability and convergence of the population-dynamics in differential evolution," *AI Communications*, vol. 22, no. 1, pp. 1–20, 2009.
- [24] L. Wang and F. Z. Huang, "Parameter analysis based on stochastic model for differential evolution algorithm," *Applied Mathematics and Computation*, vol. 217, no. 7, pp. 3263–3273, 2010.
- [25] A. E. Eiben and G. Rudolph, "Theory of evolutionary algorithms: a bird's eye view," *Theoretical Computer Science*, vol. 229, no. 1-2, pp. 3–9, 1999.
- [26] F. Xue, A. C. Sanderson, and R. J. Graves, "Modeling and convergence analysis of a continuous multi-objective differential evolution algorithm," in *Proceedings of the IEEE Congress on Evolutionary Computation (IEEE CEC '05)*, pp. 228–235, IEEE Press, Edinburgh, UK, September 2005.
- [27] F. Xue, A. C. Sanderson, and R. J. Graves, "Multiobjective differential evolution: algorithm, convergence analysis, and applications," in *Proceedings of the IEEE Congress on Evolutionary Computation*, vol. 1, pp. 743–745, Edinburgh, UK, September 2005.
- [28] Y. T. Zhao, J. Wang, and Y. Song, "An improved differential evolution to continuous domains and its convergence," in *Proceedings of the 1st ACM/SIGEVO Summit on Genetic and Evolutionary Computation (GEC '09)*, pp. 1061–1064, June 2009.
- [29] C. F. Sun, *Differential evolution and its application on the optimal scheduling of electrical power system [Ph.D. thesis]*, Huazhong University of Science and Technology, 2009 (Chinese).
- [30] Y. C. He, X. Z. Wang, K. Q. Liu, and Y. Q. Wang, "Convergent analysis and algorithmic improvement of differential evolution," *Journal of Software*, vol. 21, no. 5, pp. 875–885, 2010 (Chinese).
- [31] S. Ghosh, S. Das, A. V. Vasilakos, and K. Suresh, "On convergence of differential evolution over a class of continuous functions with unique global optimum," *IEEE Transactions on Systems, Man, and Cybernetics B*, vol. 42, no. 1, pp. 107–124, 2012.
- [32] Z. Zhan and J. Zhang, "Enhance differential evolution with random walk," in *Proceedings of the 14th International Conference on Genetic and Evolutionary Computation Conference Companion (GECCO '12)*, pp. 1513–1514, 2012.
- [33] Z. B. Hu, Q. H. Su, S. W. Xiong, and F. G. Hu, "Self-adaptive hybrid differential evolution with simulated annealing algorithm for numerical optimization," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '08)*, pp. 1189–1194, Hong Kong, China, June 2008.

- [34] A. Ghosh, S. Das, A. Chowdhury, and R. Giri, "An improved differential evolution algorithm with fitness-based adaptation of the control parameters," *Information Sciences*, vol. 181, no. 18, pp. 3749–3765, 2011.
- [35] C. Z. Chen, F. L. Jin, X. Y. Zhu, and G. Z. Ouyan, *Mathematics Analysis*, Higher Education Press, 2000.
- [36] C. J. F. T. Braak, "A Markov chain Monte Carlo version of the genetic algorithm differential evolution: easy Bayesian computing for real parameter spaces," *Statistics and Computing*, vol. 16, no. 3, pp. 239–249, 2006.
- [37] J. He and X. H. Yu, "Conditions for the convergence of evolutionary algorithms," *Journal of Systems Architecture*, vol. 47, no. 6, pp. 601–612, 2001.
- [38] G. Rudolph, "Convergence of evolutionary algorithms in general search spaces," in *Proceedings of the IEEE International Conference on Evolutionary Computation (ICEC '96)*, pp. 50–54, May 1996.
- [39] Z. Michalewicz, "Evolutionary algorithms for constrained parameter optimization problems," *Evolutionary Computation*, vol. 4, no. 1, pp. 1–32, 1996.
- [40] L. Eshelman and J. Schaffer, "Real-coded genetic algorithms and interval-schemata," in *Foundations of Genetic Algorithms-2*, L. Whitley, Ed., pp. 187–202, Morgan Kaufmann, San Mateo, Calif, USA, 1993.
- [41] T. Bäck, F. Hoffmeister, and H. P. Schwefel, "A survey of evolution strategies," in *Proceedings of the 4th International Conference on Genetic Algorithms*, pp. 2–9, 1991.
- [42] D. B. Fogel, *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*, IEEE Press, Piscataway, NJ, USA, 1995.
- [43] X. S. Wang, M. L. Hao, and Y. H. Cheng, "On the use of differential evolution for forward kinematics of parallel manipulators," *Applied Mathematics and Computation*, vol. 205, no. 2, pp. 760–769, 2008.
- [44] C. E. Chiang, W. P. Lee, and J. S. Heh, "A 2-Opt based differential evolution for global optimization," *Applied Soft Computing Journal*, vol. 10, no. 4, pp. 1200–1207, 2010.
- [45] J. Y. Yan, Q. Ling, and D. M. Sun, "A differential evolution with simulated annealing updating method," in *Proceedings of the International Conference on Machine Learning and Cybernetics*, pp. 2103–2106, Dalian, China, August 2006.
- [46] F. Neri and V. Tirronen, "Recent advances in differential evolution: a survey and experimental analysis," *Artificial Intelligence Review*, vol. 33, no. 1-2, pp. 61–106, 2010.
- [47] J. Brest, S. Greiner, B. Bošković, M. Mernik, and V. Zumer, "Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 6, pp. 646–657, 2006.
- [48] Y. Wang, Z. X. Cai, and Q. F. Zhang, "Differential evolution with composite trial vector generation strategies and control parameters," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, pp. 55–66, 2011.
- [49] J. Derrac, S. García, D. Molina, and F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 3–18, 2011.
- [50] D. E. Goldberg and P. Segrest, "Finite markov chain analysis of genetic algorithm," in *Proceedings of the International Conference on Genetic Algorithms*, Hillsdale, NJ, USA, 1987.
- [51] J. Suzuki, "A Markov chain analysis on simple genetic algorithms," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 25, no. 4, pp. 655–659, 1995.
- [52] L. Ming, Y. P. Wang, and Y. M. Cheng, "On convergence rate of a class of genetic algorithms," in *Proceedings of the World Automation Congress*, Budapest, Hungary, 2006.