International Journal of Hybrid Information Technology
Vol. 9, No.7 (2016), pp. 191-206
http://dx.doi.org/10.14257/ijhit.2016.9.7.18

Convergent Stochastic Differential Evolution Algorithms

Liang Sun[1], Hongwei Ge[1] and Limin Wang[2]

[1]*College of Computer Science and Technology, Dalian University of Technology, Dalian 116023, China*
[2] *Department of Management Science and Information Engineering, Jilin University of Finance and Economics, Changchun 130117, China*
liangsun@dlut.edu.cn, hwge@dlut.edu.cn, wlm_new@163.com

Abstract

Differential evolution (DE) algorithms have been extensively and frequently applied to solve optimizationproblems. Theoretical analyses of their properties are important to understand the underlying mechanismsand to develop more efficient algorithms. In this paper, firstly, we introduce an absorbing Markovsequence to model a DE algorithm. Secondly, we propose and prove two theorems that provide sufficientconditions for DE algorithm to guarantee converging to the global optimality region. Finally, we design two DE algorithms that satisfy the preconditions of the two theorems, respectively. The two proposed algorithmsare tested on the CEC2013 benchmark functions, and compared with other existing algorithms.Numerical simulations illustrate the converge, effectiveness and usefulness of the proposed algorithms.

Keywords*: Differential evolution, Absorbing Markov sequence, Global search, Local search, Convergent.*

1. Introduction

The differential evolution (DE) algorithm is a competitiveevolutionary computing technique for solvingoptimization problems. It is simple in concept, easy in implementation, fast in searching, and doesnot require specific domain knowledge. Since it isfirst introduced by Storn and Price [1,2], it has attractedthe attention of researchers and has obtained superiorperformance on benchmark functions and realworldproblems [3].

Eventhough DE has become a popular optimizationalgorithm and much progress has been made, there still exist a big room for algorithm to improveits performance through deeper empirical and theoreticalstudies. The empirical studies provide aDE researcher an empirical way to improve performance.Some studies have been concentrated on thetuning of the control parameters such as the mutationscale factor *F*, crossover rate*CR*, and populationsize *NP* [4-6]. Some studies have been concentrated ondesigning new mutation and crossover operators sothat the exploration and exploitation dilemma of thealgorithm are suitably balanced [7-9].

On the other hand, theoretical studies have alsobeen conducted to obtain better understanding of a DE's execution process and to guide better designandimplementation. Some studies have been concentrated on the analysis of the population variance, crossover, and search dynamics of DE. For example,Zaharia et al. analyzed the impact on the expectedpopulation mean and variance of different mutationand crossover operators [10], and they extended theirwork to analyze the influence of the crossover rateon the distribution of the number of mutated componentsand on the probability for a component to betaken from the mutant vector [11]. Dasgupta et al. proposeda simple mathematical model

ISSN: 1738-9968 IJHIT

of the underlyingevolutionary dynamics of a 1-D DE-population,and the objective is to study the evolutionary searchdynamics in DE [12]. Some studies have been concentratedon the analysis of the time complexity, andthe convergence of DE. For example, Zielinski et al. investigated the runtime complexity of DE forvarious stopping criteria [13]. Xue et al. performed the mathematical modeling and convergence analysisof continuous multiobjective DE under simplifiedassumptions [14].

The above studies facilitated the empirical andtheoretical study of DE. However, in the design ofDE, there are still some issues not solved, which, ifsuitably addressed, can lead to more robust and efficientalgorithms. As far as DE concerned, how tomake the algorithm guarantee converge to an optimalsolution is still an unanswered issue. In this paper,to address this issue, we present and prove twotheorems which provide sufficient conditions for researchersto design new algorithms that are guaranteedto converge to an optimality solution. Thisis achieved in three steps. In the first step, we introduce the absorbing Markov sequence model intoDE methodology. In the second step, based on Solisand Wets's convergence proofs for random searchalgorithms15, we propose and prove two theoremsfollowed by a modified DE model. Finally, based onthe proposed model, we derive two DE algorithms, namely StDE-C and StDE-G. The StDE-C and theStDE-G can satisfy the preconditions of the two algorithms, respectively. Thus they can converge tothe global optimality region. The framework is formulatedin such a way so that it gives deeper insightinto the execution process of DE as well as providesresearchers with ways to improve their algorithms'performance.

The remainder of this paper is organized as follows.Section 2 describes the main concepts of DE.Section 3 models the DE using an absorbing Markovsequence. Section 4 proposes and proves two theoremsthat guarantee DE convergence. In Section 5,a modified stochastic DE model is proposed, withwhich two DE algorithms are designed. Section 6describes the test functions, and provides simulationresults for illustration and comparison, followed byconcluding remarks in Section 7.

2. Basic Concepts and Formulations of Differential Evolution

Given a solution space $S = [b_{1l}, b_{1u}] \times [b_{2l}, b_{2u}] \times \cdots \times [b_{nl}, b_{nu}] \subseteq R^n$, and an objective functionf: $S \to R$. DE for minimal optimization problemscan be described as: find $\vec{x}^* \in S$, so that $\forall \vec{x} \in S, f(\vec{x}^*) \leq f(\vec{x})$. In a DE system, the candidate solution,also referred to as chromosome, is representedby an n dimensional vector:

$$\vec{x}_{i,g} = [x_{1,ig}, x_{2,ig}, \cdots, x_{n,ig}], \forall i \in \{1,2, \cdots, NP\}, \quad (1)$$

Where *NP* is the population size, g is the g-th generationof the population. Initially, the *NP* chromosomesare generated randomly. For optimization, thealgorithm works through a cycle of stages of mutation, crossover, and selection, which can be describedas follows:

1. Mutation: in the mutation process, the algorithm creates a donor chromosome $\vec{v}_{ij}$ for each target $\vec{x}_{ij}$. The frequency used mutation operators are as follows:

a. "DE/best/1"
$$\vec{v}_{i,g} = \vec{x}_{best,g} + F(\vec{x}_{r_1,g} - \vec{x}_{r_2,g}); \quad (2)$$
b. "DE/rand/1"
$$\vec{v}_{i,g} = \vec{x}_{r_1,g} + F(\vec{x}_{r_2,g} - \vec{x}_{r_3,g}); \quad (3)$$
c. "DE/current-to-best/1"
$$\vec{v}_{i,g} = \vec{x}_{i,g} + F(\vec{x}_{best,g} - \vec{x}_{i,g}) + F(\vec{x}_{r_1,g} - \vec{x}_{r_2,g}); \quad (4)$$
d. "DE/best/2"

$$\vec{v}_{i,g} = \vec{x}_{best,g} + F(\vec{x}_{r_1,g} - \vec{x}_{r_2,g}) + F(\vec{x}_{r_3,g} - \vec{x}_{r_4,g}); \quad (5)$$

e. "DE/rand/2"

$$\vec{v}_{i,g} = \vec{x}_{r_1,g} + F(\vec{x}_{r_2,g} - \vec{x}_{r_3,g}) + F(\vec{x}_{r_4,g} - \vec{x}_{r_5,g}); \quad (6)$$

f. "DE/current-to-best/2"

$$\vec{v}_{i,g} = \vec{x}_{i,g} + F(\vec{x}_{best,g} - \vec{x}_{i,g}) + F(\vec{x}_{r_1,g} - \vec{x}_{r_2,g}) + F(\vec{x}_{r_3,g} - \vec{x}_{r_4,g}); \quad (7)$$

where$\vec{x}_{best,g}$ is the best ever found individual at the$g$-th generation, r_1, r_2, r_3, r_4,r_5 are exclusive integers and are not equal to i, and$F$>0 is the control parameter for scaling the differentialvariation.

2. Crossover: in the crossover process, the donorchromosome $\vec{v}_{i,g}$ mixes its components with the targetchromosome$\vec{x}_{i,g}$ to construct a trial chromosome$\vec{u}_{i,g}$. The formulation can be described as:

$$u_{j,ig} = \begin{cases} v_{j,ig} & if\ rand_{i,j} \leq CR\ or\ j = j_{rand} \\ x_{j,ig} & otherwise \end{cases} \quad (8)$$

where $rand_{i,j}$ is a uniformly distributed randomnumber in the interval [0,1], $\mathrm{CR} \in (0,1)$ is the crossover rate, and $j_{rand} \in \{1,2,\cdots n\}$ is a randomly selected indice which ensures that the trial chromosome $u_{j,g}$ is not identical to the target chromosome $\vec{x}_{i,g}$.

3. Selection: the selection process determines whether the target chromosome $\vec{x}_{i,g}$ or the trial chromosome $u_{j,g}$ survives to the next generation. Theformulation can be described as:

$$\vec{x}_{i,g+1} = \begin{cases} \vec{u}_{i,g} & if\ f(\vec{u}_{i,g}) \leq f(\vec{x}_{i,g}) \\ \vec{x}_{i,g} & if\ f(\vec{u}_{i,g}) > f(\vec{x}_{i,g}) \end{cases} \quad (9)$$

It can be seen from Eq.(9) that the chromosomeyielding a better objective function value will survive.

The procedure of a standard DE for optimizationproblem is as follows, and almost all variants of DEalgorithms follow the similar procedure [1, 2].

1. Set population size NP, and initialize the NPchromosomes; Set generation g=1, and identify the best chromosome$\vec{x}_{best,g}$ in the population.
2. For the current population, conduct the designatedmutation, crossover, and selection operators.
3. Set current generation g=g+1, identify thebest chromosome in the population $\overrightarrow{xbest}$, and update the best ever found chromosome accordingto:

$$\vec{x}_{best,g} = argmin\{f(\overrightarrow{xbest}),\quad f(\vec{x}_{best,g-1})\} \quad (10)$$

4. If the stopping criterion is met, output $\vec{x}_{best,g}$ and its objective value; otherwise, go back toStep 2.

3. Modeling DE Using Absorbing Markov Sequence

For a minimal optimization problem with n variables,each dimension of the n dimensional solutionspace S represents one variable to be optimized. Eachchromosome can be considered as a discrete point in S, and all chromosome's information at time t can beconsidered as the state of the DE. More specifically, at time t (generation t), let the state of all the chromosomebe $pop(t)$, the state of the best ever foundchromosome be g$best(t)$. Then the execution processof DE can be described by a random process $\{\mathcal{E}(t)\}_{t=0}^{+\infty}$

where$\mathcal{E}(t) = \{pop(t), gbest(t)\}$ denotes the state of the DE at time *t*. Let *E* be the state space composed by all $\mathcal{E}$. Consider random process$\{\mathcal{E}(t)\}_{t=0}^{+\infty}$, the following lemma is true:

Lemma 1. Random process $\{\mathcal{E}(t)\}_{t=0}^{+\infty}$ of the DE has Markov property.

Proof. Because random process $\{\mathcal{E}(t)\}_{t=0}^{+\infty}$ of DE is time discrete, $\{\mathcal{E}(t)\}_{t=0}^{+\infty}$ can be represented as $\{\varepsilon(0), \varepsilon(1), \varepsilon(2), \cdots\}$. According to steps from 2 to 4 described in Section 2, the *pop(t)* is updated by the mutation, crossover, selection process of the DE algorithm, and the *gbest(t)* is updated by Eq.(10). Since the states *pop*(*t*−1), *gbest*(*t*−1) are known at time *t*, therefore, for any $E' \subseteq E$:

$$P\{\mathcal{E}(t) \in E' | \mathcal{E}(t-1), \mathcal{E}(t-2), \cdots, \mathcal{E}(0)\} = P\{\mathcal{E}(t) \in E' | \mathcal{E}(t-1)\}$$

which means that $\{\mathcal{E}(t)\}_{t=0}^{+\infty}$ has Markov property. Because $\{\mathcal{E}(t)\}_{t=0}^{+\infty}$ is a process with discrete time, it is a Markov sequence. This completes the proof.■

For a minimal optimization problem, the optimal solution $\vec{x}^* \in S$ satisfies that $\forall \vec{x} \in S,\ f(\vec{x}^*) \leq f(\vec{x})$. The solution sequence of DE can be described by $\{gbest(t)\}_{t=0}^{+\infty}$. Therefore the DE convergence means that the limit of sequence $\{f(gbest(t))\}_{t=0}^{+\infty}$ exists, and satisfies:

$$\lim_{x \to \infty} f\big(gbest(t)\big) = f(\vec{x}^*) \quad (11)$$

However, for an optimization problem, there are some pathological situations such as when the optimal solution $\vec{x}^*$ occurs at a point where $f(\mathrm{x}) = x^2$ when $\mathrm{x} \neq 1$ and $f(1) = -10$. Thus the true optimal solution will never be discovered unless the algorithm specifically tests $\mathrm{x} = 1$. Thus lead to replacing the search for the infimum with the essential infimum of *f* on *S*, which is defined as follows:

$$\lim_{x \to \infty} f\big(gbest(t)\big) = f(\vec{x}^*)\alpha = \inf\{u : v(\{x \in S | f(x) < u\}) > 0\} \quad (12)$$

Where *v*(*A*) is the Lebesgue measure of set *A*. Because *gbest*(*t*) is a discrete point in n dimensional solution space *S*, *v*({*gbest(t)*})=0, therefore *P*{*gbest*(*t*)=$\vec{x}^*$}=0. It means that the DE can not exactly find the true optimal solution $\vec{x}^*$. Hence the DE seeks to establish convergence to a region surrounding the essential infimum α. Thus the optimal region should be defined as follows.

Definition 1. A region $R_{\varepsilon,M}$ is said to be an optimal region, if

$$R_{\varepsilon,M} = \begin{cases} \{x \in S | f(x) < \alpha + \varepsilon\} & \alpha \ is \ finite, \\ \{x \in S | f(x) < M\} & \alpha < -\infty \end{cases} \quad (13)$$

Where $\varepsilon > 0$ and *M*<0.

In this section, we will describe DE as an absorbing Markov model. Let's first give the definition of an optimal state space.

Definition 2. A state space $E^* \subset E$ is said to be an optimal state space if and only if $\forall$*ε*(*t*)=(*pop*(*t*), *gbest*(*t*))$\in E^*$, satisfies that *gbest*(*t*)$\in R_{\varepsilon,M}$.

It can be seen from Definition 2 that $E^*$is the objective state space of DE. Given a random sequence $\{\mathcal{E}(t)\}_{t=0}^{+\infty}$of DE and an optimal state space E^*, the following lemma is true:

Lemma 2. Random process $\{\mathcal{E}(t)\}_{t=0}^{+\infty}$ of DE is an absorbing Markov sequence.

Proof. According to Lemma 1, $\{\mathcal{E}(t)\}_{t=0}^{+\infty}$ is a Markov random sequence. At time t, if $\mathcal{E}(t) \in E^*$, according to Definition 2, $gbest(t) \in R_{\varepsilon,M}$. According to Step 3 of DE procedure, gbest(t) is updated by Eq.(10). Thus *f*(*gbest*(*t*+1))<*f*(*gbest*(*t*)) is satisfied, then *gbest*(*t*+1)$\in R_{\varepsilon,M}$, it means that $\mathcal{E}(t+1) \in E^*$. As a result, $P\{\mathcal{E}(t+1) \in E^*\} = 1$, so $\{\mathcal{E}(t)\}_{t=0}^{+\infty}$ is an absorbing Markov sequence. This completes the proof. ■

4. Sufficient Conditions for DE Guaranteed Convergence

In this section, we will propose two theorems that provide sufficient conditions for DE algorithm to be guaranteed to converge to the optimality region. Therefore the two theorems will provide guideline to better design and implement DEs. The first theorem guarantees that DE converges to the optimality region as a global search method. The second theorem guarantees that DE converges to the optimality region as a local search method. These two theorems are based on Solis and Wets's paper which provides convergence proofs for random search techniques [15].

4.1. Global search and Local Search Methods

Solis and Wets distinguished ideas between global and local search methods based on the properties of the sequence of probability measures μ_k utilized.

Definition 3. Suppose$\{\mu_k\}$isthesequenceof probability measures corresponding to a method. The method is said to be a global search method if for any (Borel) subset *A* of *S* with *v*(*A*)>0, satisfies:

$$\prod_{k=0}^{+\infty}\left(1-\mu_k(A)\right)=0$$

where$\mu_k(A)$ is the probability of subset A being generated by μ_k.

Definition 4. Suppose μ_k isis the sequence of probability measures corresponding to a method, let $M_k$be the support set of μ_k. The method is said to be a local search method if it has the μ_k with M_k for all *k*, except for a finite number of k values, satisfies *v*(*S*∩M_k)<*v*(*S*).

According to *Definitions* 3and 4 above, global search method can be explained as follows: given any subset *A* of *S* with positive "volume", when a global search method generating random samples, the probability for it repeatedly miss set *A* is zero. For a local search method, the support set M_k of μ_k is bounded and *v*(*S*∩μ_k)<*v*(*S*). So there may exists a set M′, satisfying v(M′)>0 and M′∩($\cup_{k=0}^{+\infty} M_k$)= Ø , thus $\mu_0(M')=\mu_1(M')=\cdots=0$. Thatmeansthat there may exist some regions with positive "volume" that will never be searched.

In a DE system, μ_{ik} is the probability measure corresponding to chromosome *i* at time *k*in the solution space *S*, and M_{ik} is the support set of μ_{ik}. $\mu_k$is the probability measure corresponding to all the chromosomes at time *k* in the solution space *S*, and $M_k=\cup_{i=1}^{+\infty} M_{i,k}$ is the support set of μ_{ik}. In the following sections, for the purpose of convenience, we call M_{ik} the support set of chromosome *i* at time *k*, and M_k the support set of the whole population.

4.2. DE Convergence as a Global Search Method

In this subsection, we will propose a theorem that provides sufficient condition for DE to be guaranteed to converge as a global search method.

Theorem 3. Given a random process $\{\mathcal{E}(t)\}_{t=0}^{+\infty}$of DE, and an optimality state space *E* . Let M_{ik}be the support set of chromosome *i* at time *k*, and $M_k=\cup_{i=1}^{NP} M_{i,k}$ be the support set of the whole population. If$M_k=S$forall$k\geq 0$,then:

$$\lim_{t\to\infty} P\{\mathcal{E}(t)\in E^*\}=1 \tag{11}$$

where NP is the number of chromosomes in the population.

Theorem 1 can be explained intuitively as follows. Because $\mu_k=\cup_{i=1}^{m} M_{i,k}=S$, thus$R_{\varepsilon,M}\subset M_k$. Because $v(R_{\varepsilon,M})>0$ the probability of repeatedly missing optimality region $R_{\varepsilon,M}$, when generating new chromosomes, is zero. With the fact that DE has

absorbing Markov sequence property, the DE will guarantee converging to the optimality region.

According to *Theorem* 1, in order to guarantee DE converging to the optimality region, one way is to make the support set of the whole population M_k covers the entire solution space *S* at each time step.

4.3. DE Convergence as a Local Search Method

In this subsection, we will propose the second theorem that provides sufficient condition for DE to be guaranteed to converge as a local search method.

Theorem 4. Given a random process $\{\mathcal{E}(t)\}_{t=0}^{+\infty}$of DE, and an optimality state space E^*. Let $M_{i,k}$ be the support set of chromosome*i*at time *k*, and $M_k = \cup_{i=1}^{NP} M_{i,k}$ be the support set of the whole population. If there exist $\gamma > 0$ and $\mu > 0$ for all $k \geq 1$, at least one of the following is satisfied:

1. Set $B_k = \{x | dist(x, R_{\varepsilon,M}) < dist(gbest(k-1), R_{\varepsilon,M}) - \gamma\}$ satisfies that $v(M_k \cap B_k) > u$;
2. $v(M_k \cap R_{\varepsilon,M}) > u$

It can be seen that for a DE at each iteration step k, if the "volume" of $M_k \cap (B_k \cup R_{\varepsilon,M})$is larger than zero, with the fact that DE has absorbing Markov sequence property, the the DE can guarantee converging to the optimality region. For the purpose of visualization, Figure 1

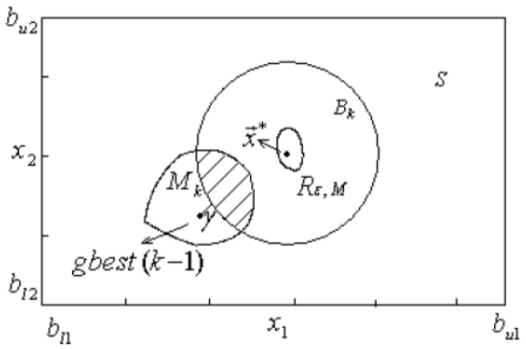

Figure 1. Relationship of M_k, B_k, and $R_{\varepsilon,M}$, case: $(M_k \cap B_k) > u$.

shows an example of the relationship among M_k, B_k, and $R_{\varepsilon,M}$for a 2-dimensional optimization problem, where the shading area is $M_k \cap B_k$.

According to Theorem 2, in order to guarantee DE to converge to the optimality solution space, another way is to make the support set M_k of the whole population covers the area that is "closer" to $R_{\varepsilon,M}$ at each time step.

5. Stochastic Differential Evolution Algorithms

As aforementioned, we have two theorems that guarantee DE converging to the optimality region when the conditions in either theorem are satisfied. The first theorem requires having the support set of the whole population covering the entire solution space. The second theorem requires having the support set of the whole swarm covering the area that is "closer" to the optimality region. In this section, we will design a stochastic differential evolution model according to the two theorems, which results in two StDE algorithms, respectively. The first StDE, namely StDE-C, is designed to satisfy the

precondition of *Theorem* 1, while the other StDE, namely StDE-G, is designed to satisfy the precondition of *Theorem* 2.

5.1. Stochastic Differential Evolution Optimizer

In the following, we will design the StDE model. The exclusive features of the StDE model include:

1. Each chromosome*i*representsastochastic region that is described by a specific distribution function, but not represents a point in the solution space. It uses its chromosome vector $\vec{x}_i = [x_{i1}, x_{i1}, \cdots, x_{in}]$, to represent the center of the stochastic region.
2. After the operators of mutation, crossover, and selection are conducted, a proportion of the chromosomes undergo a sampling process. The sampling process is achieved by firstly generating a new chromosome vector according to the distribution function, and replacing the old chromosome vector with the new generated one, regardless of the objective function value deterioration.
3. The parameters that are used to describe the distribution function are dynamically adjusted.

Based on the above features, the procedure of StDE model for optimization is as follows.

1. Set population size *NP*, and initialize the *NP* chromosomes; Set generation *g*=1, and identify the best chromosome$\vec{x}_{best,g}$in the population; Initialize the parameters that describe the stochastic regions of each chromosome; Set the parameter *c*(0<c<1) that represents the sampling rate;
2. For the current population, conduct the designated mutation, crossover, and selection operators.
3. Set current generation *g*= *g*+1, identify the best chromosome in the population, and update the best ever found chromosome according to Eq.(10).
4. For each chromosome *i*, generate a random real number *r* in the interval [0,1], if *r*<c, generate a new chromosome $\vec{x'}_i$ in its corresponding stochastic region, replace $\vec{x}_i$with $\vec{x'}_i$; update the parameters that describe the stochastic region.
5. If the stopping criterion is met, output $\vec{x}_{best,g}$ and its objective value; otherwise, go back to Step 2.

We derive two StDE algorithms using different types of stochastic region. In the first StDE algorithm, the so calledStDE-C, the stochastic region is described by Cauchy distribution. Thus, the stochastic region of chromosome *i*is represented by $[C(x_{i1}, t_{i1}), C(x_{i2}, t_{i2}), \cdots, C(x_{in}, t_{in})]$, where $\vec{x}_i = [x_{i1}, x_{i1}, \cdots, x_{in}]$ and $\vec{t}_i = [t_{i1}, t_{i1}, \cdots, t_{in}]$ are position vector and scale vector of the Cauchy distribution, respectively. In the second StDE algorithm, the so calledStDE-G, the stochastic region is described by Gaussian distribution. Thus, the stochastic region of chromosome i is represented by $[N(x_{i1}, \sigma_{i1}), N(x_{i2}, \sigma_{i2}), \cdots, N(x_{in}, \sigma_{in})]$, where $\vec{x}_i = [x_{i1}, x_{i1}, \cdots, x_{in}]$ and $\vec{\sigma}_i = [\sigma_{i1}, \sigma_{i1}, \cdots, \sigma_{in}]$are position vector and variance vector of the Gaussian distribution, respectively.

As mentioned above, the parameters that describe the stochastic region are adjusted during the execution process. The rules of how to adjust these parameters are designed empirically. In StDE-C, at the *k*-th iteration, the following rule is adopted:

$$t_{ij}(k) = e^{-k/T_1} \tag{14}$$

In StDE-G, at the *k*-th iteration, the following rule is adopted:

$$\sigma_{ij}(k) = \frac{1}{20} e^{-k/T_2} \cdot \left(b_{ju} - b_{jl}\right) + \varepsilon, \tag{15}$$

whereT_1 and T_2 are positive integers, b_{jl} and b_{ju} are the lower and upper boundaries of the j-th dimension, and $\varepsilon > 0$ is a real number.

To further improve the performance of the StDE-C and StDE-G, the cross-cluster mutation pro- posed in our previous work in Ref. 17 is incorporated. The cross-cluster mutation is designed to enhance exploitation and exploration ability of the algorithm. Firstly, the operators in Eqs.(2)-(6) are identified as exploitation-biased or exploration-biased according to the Hopkins test value (h-value). Following this, the population of the entire population is divided into subpopulations. For the chromosomes taken from the same subpopulation, the exploitation-biased operators are applied, and for the chromosomes taken from different subpopulation, the exploration-biased operators are applied.

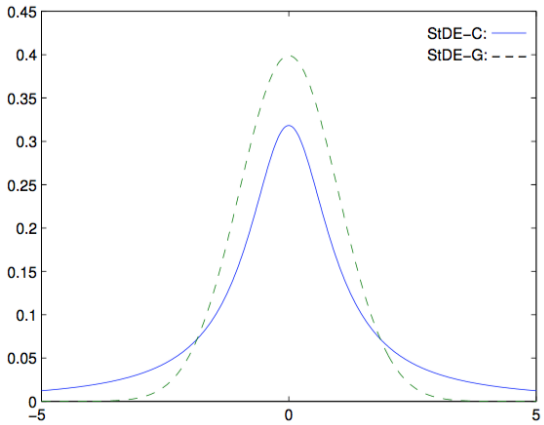

Figure 2. Comparison between Cauchy and Gaussian Density Functions

5.2. Convergence Analysis

According to Step 4 of the StDE model, each chromosome undergoes a sampling process with a certain probability. It means that its support set is further expanded to its corresponding stochastic region. Let's examine the properties of the stochastic region of StDE-C and StDE-G, respectively. The one dimensional Cauchy density function centered at zero is defined by:

$$p(x) = \frac{1}{\pi} \cdot \frac{t}{t^2 + x^2}, -\infty < x < +\infty \quad (16)$$

where$t$>0 is the scale parameter. The corresponding distribution function is:

$$F(x) = \frac{1}{2} + \frac{t}{\pi} \arctan\left(\frac{x}{t}\right), -\infty < x < +\infty \quad (17)$$

The one dimensional Gaussian density function centered at zero is defined by:

$$p(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{x^2}{2\sigma^2}}, -\infty < x < +\infty \quad (16)$$

where σ>0 is the variance parameter. The corresponding distribution function is:

$$F(x) = \int_{-\infty}^{x} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}} d_x, -\infty < x < +\infty \quad (17)$$

The Cauchy density function and Gaussian density function are of the same shape but not identical to each other. Figure 2 shows the difference between them by setting$t$=1andσ=1.

It can be seen from Figure .2 that Cauchy distributed region is more likely to generate a new position further away from its current center due to its long flat trails. On the other hand, the Gaussian distributed region has a higher probability to generate a new position

around its current center, and the probabilities to generate new positions in intervals [−σ,σ], [−1.98σ, 1.98σ] and 2.58σ , 2.58σ are 68.27%, 95% and 99%, respectively.

As analyzed above, because the support set of StDE-C covers the entire solution space, thus StDE-C is a global search method. According to *Theorem* 1, StDE-C can guarantee chromosomes converging to the optimality region. On the other hand, although the support set of StDE-G covers the entire solution, it has a much lower probability of generating new position beyond the interval [−2.58σ , 2.58σ] at each dimension, so the StDE-G can be

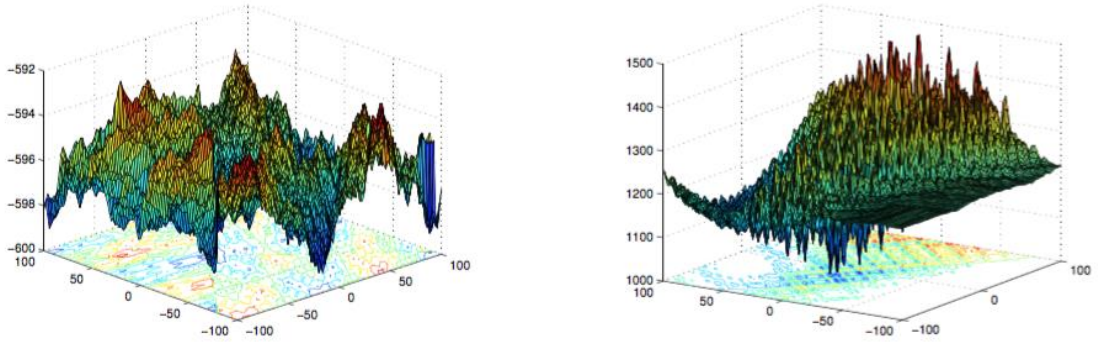

Figure 3. Surface Landscapes of F9 And F24.

regarded as a local search method. However, for the chromosome j whose position denotes the whole population's best position $\vec{p_g}$, its support set $M_{i,k}$ is $[N(p_{g1}, \sigma_{i1}), N(p_{g2}, \sigma_{i2}), \cdots, N(p_{gn}, \sigma_{in})]$. According to Eq.(15), the variance is larger than ε, then $v(M_k \cap B_k) > u$ is satisfied. According to Theorem 2, the StDE-G can guarantee chromosomes converging to the optimality region.

6. Simulation and Discussions

6.1. Test Functions and Experimental Settings

In the experiment, we select the CEC2013 as benchmarks, which include the shifted, rotated, expanded, and combined variants of the basic functions. The CEC2013 has 28 functions, the properties and formulas of them are reported in Ref. 16. Among the functions, 5 of them are unimodal, 15 of them are multimodal, and 8 of them are composition functions. As an example, Figure 3 shows the surface landscapes of f_9 and f_{24} on the first and second dimensions. They are designed to test the performance of algorithms on moderate scale optimization problems. While testing, the explicit equations of the problems are not allowed being used by the algorithms. Generally, for unimodal functions, the convergence speeds are of more interesting than the final results of optimization. On the contrary, for multimodal and composition functions, the quality of final results is much more important since they reflect an algorithm's ability of escaping from pseudo- optima and locating a good near global optimum.

The experiments are carried out following the instructions reported in the literature associated to the CEC2013 [16]. The dimensions of the functions are taken as 30 and 50, respectively. Throughout the experiments, the number of function evaluations (FEs) is used to measure the computational efforts. For each function, the maximum number of FEs is 3 × 105 for 30 dimensional problems, and 5×10^5 for 50 dimensional problems. The results are presented in terms of function error value $err_i = f_i(\vec{x}) - f_i(\overrightarrow{x^*})$, where $f_i(\vec{x})$ denotes the objective function value of chromosome $\vec{x}$, and $f_i(\overrightarrow{x^*})$ denotes the global optimum value of the function. The algorithm terminates when the maximum number of

FEs is reached or the error value is smaller than 10^{-8}. To test the stability of the algorithm, the experiment on each function is repeated for 51 times, and the best, worst, median values, the mean value, and the standard derivation of the 51 runs are presented. In the experiment, the involved parameters are selected empirically so that the best computation results having high quality solution can be obtained.

Table 1. Results of StDE-C and StDE-G (Dim=30)

Func	StDE-C					StDE-G				
	Best	Worst	Median	Mean	Std	Best	Worst	Median	Mean	Std
f_1	1.00e-08	1.00e-08	1.00e-08	1.00e-08	0.00e-00	1.00e-08	1.00e-08	1.00e-08	1.00e-08	0.00e-00
f_2	1.00e-08	1.00e-08	1.00e-08	1.00e-08	0.00e-00	1.00e-08	1.00e-08	1.00e-08	1.00e-08	0.00e-00
f_3	1.00e-08	1.00e-08	1.00e-08	1.00e-08	0.00e-00	1.00e-08	1.00e-08	1.00e-08	1.00e-08	0.00e-00
f_4	1.00e-08	1.00e-08	1.00e-08	1.00e-08	0.00e-00	1.00e-08	1.00e-08	1.00e-08	1.00e-08	0.00e-00
f_5	1.00e-08	1.00e-08	1.00e-08	1.00e-08	0.00e-00	1.00e-08	1.00e-08	1.00e-08	1.00e-08	0.00e-00
f_6	1.00e-08	1.00e-08	1.00e-08	1.00e-08	0.00e-00	1.00e-08	1.00e-08	1.00e-08	1.00e-08	0.00e-00
f_7	1.00e-08	9.42e-02	4.27e-05	4.50e-03	1.87e-02	3.50e-01	3.89e-00	1.39e-00	1.42e-00	1.10e-00
f_8	2.01e+01	2.18e+01	2.09e+01	2.09e+01	4.30e-02	2.00e+01	2.04e+01	2.03e+01	2.02e+01	8.93e-02
f_9	2.40e-00	5.12e-00	3.80e-00	4.20e-00	1.30e-00	1.00e-08	4.7e-00	1.03e-00	1.49e-00	1.40e-00
f_{10}	1.00e-08	1.00e-08	1.00e-08	1.00e-08	0.00e-00	1.00e-08	1.00e-08	1.00e-08	1.00e-08	0.00e-00
f_{11}	1.00e-08	1.00e-08	1.00e-08	1.00e-08	0.00e-00	1.30e-01	3.54e-00	1.25e-00	1.25e-00	1.53e-00
f_{12}	7.29e-01	2.73e-00	1.68e-00	1.58e-00	1.25e-00	3.51e-01	1.58e-00	9.01e-01	1.09e-00	1.14e-00
f_{13}	9.94e-01	2.85e-00	2.05e-00	2.19e-00	1.41e-00	8.15e-01	2.59e-00	1.56e-00	1.69e-00	1.91e-00
f_{14}	7.22e+01	2.99e+02	1.18e+02	1.24e+02	9.73e+01	1.58e-02	5.24e-02	3.29e-02	2.99e-02	3.65e-02
f_{15}	1.08e+02	4.99e+02	1.19e+02	2.27e+02	1.21e+02	1.52e+02	7.22e+02	3.10e+02	3.30e+02	2.45e+02
f_{16}	3.46e-02	6.35e-01	2.54e-01	2.95e-01	2.03e-01	8.95e-02	7.83e-01	1.23e-01	1.35e-01	1.06e-01
f_{17}	3.04e+01	3.58e+01	3.23e+01	3.23e+01	1.92e-00	9.83e-00	2.35e+01	1.00e+01	1.01e+01	1.12e-00
f_{18}	2.59e+01	3.02e+01	2.77e+01	2.79e+01	1.87e-00	1.15e+01	2.79e+01	1.71e+01	1.75e+01	3.76e-00
f_{19}	3.15e-01	5.57e-01	3.96e-01	4.09e-01	2.88e-01	1.20e-01	4.09e-01	2.96e-01	2.87e-01	6.04e-02
f_{20}	9.50e-01	3.02e-00	1.92e-00	1.97e-00	5.81e-01	2.14e-00	4.97e-00	2.83e-00	3.02e-00	2.10e-00
f_{21}	9.34e+01	2.45e+02	1.23e+02	1.14e+02	1.04e+02	1.12e+02	3.31e+02	1.87e+02	2.00e+02	4.59e+01
f_{22}	3.58e-00	9.56e-00	7.21e-00	7.35e-00	6.43e-00	1.45e-00	1.00e+02	7.39e-00	8.86e-00	1.94e+01
f_{23}	2.48e+02	4.29e+02	3.14e+02	3.23e+02	2.21e+02	6.20e+01	9.52e+02	4.90e+02	4.92e+02	2.88e+02
f_{24}	1.07e+01	8.79e+02	2.00e+02	2.03e+02	7.61e+01	1.03e+02	2.69e+02	2.10e+02	1.93e+02	9.84e+01
f_{25}	1.10e+02	5.42e+02	1.83e+02	1.96e+02	1.78e+01	1.46e+02	3.69e+02	2.33e+02	2.04e+02	1.65e+02
f_{26}	1.00e+02	2.00e+02	1.14e+02	1.51e+02	4.77e+01	6.45e+01	2.15e+02	1.03e+02	1.02e+02	1.14e+02
f_{27}	1.34e+02	3.70e+02	2.39e+02	2.50e+02	2.42e+02	1.65e+02	6.84e+02	3.25e+02	3.20e+02	4.08e+01
f_{28}	1.19e+02	3.24e+02	1.89e+02	2.05e+02	9.20e+01	1.67e+02	8.43e+02	2.38e+02	2.55e+02	5.53e+01

6.2. Simulation Results and Discussions

6.2.1. Results for 30-D Problems

This subsection presents the results of the StDE-C and StDE-G for the 30-D problems. The best, worst, median values, the mean value and the standard derivation of the 51 runs are presented in Table 1. It can be seen from Table 1 that both the StDE-C and StDE-G can obtain error value smaller than 10−8 for the unimodal functions f1−f5 and basic multimodal functions f6, f10. As for the rest of the 13 basic multimodal functions, the StDE-C obtains better results on 4 instances, while StDE-G obtains better results on 9 instances. As for the composition functions, the StDE-C obtains better results on 6 instances, while StDE-G obtains better results on 2 instances.

For the purpose of comparison, the top ranked algorithms in the CEC2013 competition are selected, which include the NBIPOPaCMA [18] ,icmaesils [19] and DRMA-LSCh-CMA [20]. Table 2 presents the results of the StDE-C, StDE-G, and the compared algorithms. It can be seen from Table 2 that the StDE-G obtains the best or the same best results for 18 instances, and the StDE-G obtains the best or the same best results for 15 instances. To illustrate the statistical difference between the StDE-C, StDE-G, and the compared algorithms, the Friedman test and the Holm test are conducted [21]. The results are presented in Table 3.

Table 2. Comparison Results of Stde-C And Stde-G with Other Algorithms (Dim=30).

Func	StDE-C		StDE-G		NBIPOPaCMA		icmaesils		DRMA-LSCh-CMA	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
f_1	1.00e-08	0.00e-00	1.00e-08	0.00e-00	0.00e-00	0.00e-00	1.00e-08	0.00e-00	0.00e-00	0.00e-00
f_2	1.00e-08	0.00e-00	1.00e-08	0.00e-00	0.00e-00	0.00e-00	1.00e-08	0.00e-00	0.00e-00	0.00e-00
f_3	1.00e-08	0.00e-00	1.00e-08	0.00e-00	0.00e-00	0.00e-00	1.00e-08	0.00e-00	5.91e-04	3.18e-03
f_4	1.00e-08	0.00e-00	1.00e-08	0.00e-00	0.00e-00	0.00e-00	1.00e-08	0.00e-00	0.00e-00	0.00e-00
f_5	1.00e-08	0.00e-00	1.00e-08	0.00e-00	0.00e-00	0.00e-00	1.00e-08	0.00e-00	9.35e-05	1.34e-04
f_6	1.00e-08	0.00e-00	1.00e-08	0.00e-00	0.00e-00	0.00e-00	1.00e-08	0.00e-00	0.00e-00	0.00e-00
f_7	4.50e-03	1.87e-02	1.42e-00	1.10e-00	2.31e-00	6.05e-00	7.01e-02	1.56e-01	1.54e-00	2.16e-00
f_8	2.09e+01	4.30e-02	2.02e+01	8.93e-02	2.09e+01	4.80e-02	2.09e+01	6.23e-02	2.09e+01	4.15e-02
f_9	4.20e-00	1.30e-00	1.49e-00	1.40e-00	3.30e-00	1.38e-00	4.34e-00	1.72e-00	8.79e-00	2.08e-00
f_{10}	1.00e-08	0.00e-00	1.00e-08	0.00e-00	0.00e-00	0.00e-00	1.00e-08	0.00e-00	2.56e-03	4.70e-03
f_{11}	1.00e-08	0.00e-00	1.25e-00	1.53e-00	3.04e-00	1.41e-00	2.25e-00	1.05e-00	4.17e-00	1.39e-00
f_{12}	1.58e-00	1.25e-00	1.09e-00	1.14e-00	2.91e-00	1.38e-00	1.72e-00	1.23e-00	1.38e+01	2.94e-00
f_{13}	2.19e-00	1.41e-00	1.69e-00	1.91e-00	2.78e-00	1.45e-00	2.16e-00	1.30e-00	2.84e+01	7.82e-00
f_{14}	1.24e+02	9.73e+01	2.99e-02	3.65e-02	8.10e+02	3.60e+02	7.08e+02	2.94e+02	3.12e+02	1.67e+02
f_{15}	2.27e+02	1.21e+02	3.30e+02	2.45e+02	7.65e+02	2.95e+02	2.59e+02	1.18e+02	1.56e+03	4.53e+02
f_{16}	2.95e-01	2.03e-01	1.35e-01	1.06e-01	4.40e-01	9.26e-01	3.75e-01	2.65e-01	2.10e-02	1.07e-02
f_{17}	3.23e+01	1.92e-00	1.01e+01	1.12e-00	3.44e+01	1.87e-00	3.43e+01	1.86e-00	3.86e+01	3.18e-00
f_{18}	2.79e+01	1.87e-00	1.75e+01	3.76e-00	6.23e+01	4.56e+01	4.01e+01	1.87e+01	4.38e+01	3.48e-00
f_{19}	4.09e-01	2.88e-01	2.87e-01	6.04e-02	2.23e-00	3.41e-01	2.24e-00	5.66e-01	2.01e-00	3.04e-01
f_{20}	1.97e-00	5.81e-01	3.02e-00	2.10e-00	1.29e+01	5.98e-01	1.44e+01	7.38e-01	9.70e-00	8.46e-01
f_{21}	1.14e+02	1.04e+02	2.00e+02	4.59e+01	1.92e+02	2.72e+01	1.88e+02	3.25e+01	3.02e+02	7.81e+01
f_{22}	7.35e-00	6.43e-00	8.85e-00	1.94e+01	8.38e+02	4.60e+02	5.33e+02	3.63e+02	1.91e+02	6.85e+01
f_{23}	3.23e+02	2.21e+02	4.92e+02	2.88e+02	6.67e+02	2.90e+02	2.69e+02	1.41e+02	1.38e+03	4.43e+02
f_{24}	2.03e+02	7.61e+01	1.93e+02	9.84e+01	1.62e+02	3.00e+01	2.00e+02	6.16e-04	2.03e+02	5.33e-00
f_{25}	1.96e+02	1.78e+01	2.04e+02	1.65e+02	2.20e+02	1.11e+01	2.40e+02	5.12e-00	2.36e+02	2.40e+01
f_{26}	3.00e+01	2.16e+02	3.67e+01	2.28e+02	4.77e+01	1.02e+02	1.14e+02	1.51e+02	4.77e+02	1.58e+02
f_{27}	2.50e+02	2.42e+02	3.20e+02	4.08e+01	4.69e+02	7.38e+01	3.00e+02	9.34e-03	3.64e+02	8.40e+01
f_{28}	2.05e+02	9.20e+01	2.55e+02	5.53e+01	2.69e+02	7.35e+01	2.45e+02	9.01e+01	2.96e+02	2.80e+01

Table 3. Results of The Friedman Test and Holm Test. ($\alpha = 0.05$)

Friedman test					Holm test			
Algorithm	Rank	χ^2	p value	Diff.?	Algorithm	z	p value	Diff.?
StDE-C	2.09	43.654	0.001	Yes	StDE-C *v.s.* StDE-G	0.17	0.5636	No
StDE-G	2.16							
NBIPOPaCMA	3.61				StDE-C *v.s.* NBIPOPaCMA	2.60	0.0047	Yes
					StDE-G *v.s.* NBIPOPaCMA	2.43	0.0075	Yes
icmaesils	3.07				StDE-C *v.s.* icmaesils	2.31	0.0104	Yes
					StDE-G *v.s.* icmaesils	2.15	0.0158	Yes
DRMA-LSCh-CMA	4.07				StDE-C *v.s.* DRMA-LSCh-CMA	3.68	0.0002	Yes
					StDE-G *v.s.* DRMA-LSCh-CMA	3.51	0.0003	Yes

It can be seen from the Friedman test results that the differences among the five algorithms are statistically relevant with 95% certainty. The StDE-G obtains the first rank, and the StDE-C obtains the second rank. When we compare the StDE-C and the StDE-G with each other, the Holm test result shows that the difference is not statistically relevant. The StDE-C obtains slightly better results, as indicated by p = 0.5636. When we compare the StDE-C and the StDE-G with the NBIPOPaCMA, the icmaesils, and the DRMA-LSCh-CMA, the Holm test shows that both the StDE-C and the StDE-G obtains better results, and the difference are statistically relevant with 95% certainty.

Table 4. Results of StDE-C and StDE-G (Dim=50).

Func	StDE-C					StDE-G				
	Best	Worst	Median	Mean	Std	Best	Worst	Median	Mean	Std
f_1	1.00e-08	1.00e-08	1.00e-08	1.00e-08	0.00e-00	1.00e-08	1.00e-08	1.00e-08	0.00e-00	0.00e-00
f_2	1.00e-08	1.00e-08	1.00e-08	1.00e-08	0.00e-00	1.00e-08	1.00e-08	1.00e-08	0.00e-00	0.00e-00
f_3	1.37e-02	7.53e-02	2.23e-02	2.50e-02	5.18e-02	3.19e-02	7.53e-02	5.45e-02	5.17e-02	1.33e-02
f_4	1.00e-08	1.00e-08	1.00e-08	1.00e-08	0.00e-00	1.00e-08	1.00e-08	1.00e-08	1.00e-08	0.00e-00
f_5	1.00e-08	1.00e-08	1.00e-08	1.00e-08	0.00e-00	1.00e-08	1.00e-08	1.00e-08	1.00e-08	0.00e-00
f_6	1.00e-08	1.00e-08	1.00e-08	1.00e-08	0.00e-00	1.59e+01	5.15e+01	2.76e+01	2.88e+01	9.35e-00
f_7	1.47e-01	9.84e-01	4.95e-01	5.47e-01	5.07e-01	1.94e-00	5.74e-00	3.35e-00	3.76e-00	2.59e-00
f_8	2.09e+01	2.13e+01	2.11e+01	2.11e+01	6.04e-02	2.08e+01	2.21e+01	2.12e+01	2.11e+01	6.14e-02
f_9	1.64e-00	7.42e-00	5.98e-00	5.39e-00	2.37e-00	1.13e-00	4.28e-00	2.37e-00	2.47e-00	2.96e-00
f_{10}	1.00e-08	1.00e-08	1.00e-08	1.00e-08	0.00e-00	1.00e-08	1.00e-08	1.00e-08	1.00e-08	0.00e-00
f_{11}	4.72e-00	8.09e-00	5.44e-00	5.63e-00	1.78e-00	2.14e-00	7.63e-00	5.59e-00	5.72e-00	3.27e-00
f_{12}	1.62e-00	1.09e+01	4.57e-00	4.12e-00	7.35e-00	1.98e-00	5.41e-00	3.37e-00	3.28e-00	2.35e-00
f_{13}	2.86e-00	1.12e+01	6.29e-00	7.31e-00	2.58e-00	1.36e-00	7.31e-00	4.42e-00	4.69e-00	3.51e-00
f_{14}	1.35e+02	1.24e+03	5.41e+02	6.07e+02	3.39e+02	1.84e+02	7.67e+02	3.89e+02	4.24e+02	4.59e+02
f_{15}	2.56e+02	6.11e+02	3.84e+02	3.29e+02	3.70e+02	2.64e+02	4.92e+02	3.84e+02	3.90e+02	1.84e+02
f_{16}	2.59e-01	7.04e-01	2.91e-01	3.98e-01	2.73e-01	1.45e-01	5.19e-01	3.12e-01	3.15e-01	2.21e-01
f_{17}	2.04e+01	7.11e+01	4.00e+01	4.04e+01	2.45e+01	1.73e+01	4.04e+01	3.14e+01	2.99e+01	2.53e+01
f_{18}	5.41e+01	8.33e+01	7.01e+01	7.77e+01	9.09e-00	4.37e+01	1.04e+02	6.77e+01	7.29e+01	6.33e+01
f_{19}	4.59e-00	7.76e-00	5.71e-00	5.26e-00	2.37e-01	1.84e-01	5.26e-01	3.21e-01	3.09e-01	1.35e-08
f_{20}	8.45e-00	2.00e+01	1.18e+01	1.01e+01	4.82e-01	1.73e+01	6.57e+01	4.23e+01	4.08e+01	2.83e+01
f_{21}	2.00e+02	4.26e+02	3.43e+02	3.01e+02	6.51e+01	9.75e+01	4.01e+02	2.23e+02	2.36e+02	1.46e+02
f_{22}	6.15e+01	1.23e+02	9.27e+01	9.92e+01	2.71e+01	1.13e+02	4.22e+02	2.38e+02	2.40e+02	2.36e+02
f_{23}	2.19e+02	6.73e+02	3.76e+03	3.73e+02	2.35e+02	4.74e+02	8.37e+02	6.47e+02	6.52e+02	2.84e+02
f_{24}	2.04e+02	3.63e+02	2.36e+02	2.48e+02	7.23e+01	6.75e+01	4.18e+02	1.99e+02	2.03e+02	1.81e+02
f_{25}	2.14e+02	3.38e+02	2.82e+02	2.59e+02	4.98e-00	1.82e+02	4.62e+02	2.48e+02	2.61e+02	6.75e+01
f_{26}	1.15e+02	3.12e+02	1.53e+02	1.76e+02	6.02e+01	1.51e+02	4.44e+02	1.95e+02	2.01e+02	1.14e+02
f_{27}	3.38e+02	7.11e+02	5.69e+02	5.14e+02	1.08e+02	1.44e+02	7.14e+02	3.98e+02	4.10e+02	3.81e+02
f_{28}	4.00e+02	4.00e+02	4.00e+02	4.00e+02	0.00e-00	4.00e+02	4.00e+02	4.00e+02	4.00e+02	0.00e-00

6.2.2. Results for the 50-D Problems

The subsection presents the results of the StDE-C and the StDE-G for the 50-D problems. The experiments on the 30-D problems are repeated on the 50-D problems. The best, worst, median values, the mean value and the standard derivation of the 51 runs are presented in Table 4. It can be seen from Table 4 that the StDE-C and StDE-G can obtain error value smaller than $10^{-8}$for 4 of the unimodal functions f_1, f_2, f_4, f_5, and the basic multimodal function f_{10}. As for the rest of the 14 basic multimodal functions, the StDE-C obtains better results on 6 instances, while the StDE-G obtains better results on 8 instances. As for the composition functions, the StDE-C obtains better results on 3 instances, the StDE-G obtains better results on 4 instances, and the two algorithms obtain the same result on 1 instance.

The StDE-C and the StDE-G are also com- pared with the NBIPOPaCMA, the icmaesils and the DRMA-LSCh-CMA. The comparison results are presented in Table 5. It can be seen from Table 5 that the StDE-C obtains the best or the same best results for 12 instances, and the StDE-G obtains the best or the same best results for 16 instances. The Friedman test and the Holm test results are presented in Table 6. It can be seen from the Friedman test results that the difference among the five algorithms are statistically relevant with 95% certainty. The StDE-G obtains the first rank, and the StDE-C obtains the second rank. When we compare the StDE-C and the StDE-G with each other, the Holm results result shows that the difference is not statistically relevant. The StDE-C obtains slightly better results, as indicated by $p = 0.3611$. When we compare the StDE-C and the StDE-G with the NBIPOPaCMA, the icmaesils, and the DRMA-LSCh-CMA, the Holm test shows that both the StDE-C and the StDE-G obtains better results, and the differences are statistically relevant with 95% certainty.

Table 5. Comparison Results of Stde-C and Stde-G with Other Algorithms (Dim=50)

Func	StDE-C		StDE-G		NBIPOPaCMA		icmaesils		DRMA-LSCh-CMA	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
f_1	1.00e-08	0.00e-00	1.00e-08	0.00e-00	0.00e-00	0.00e-00	1.00e-08	0.00e-00	0.00e-00	0.00e-00
f_2	1.00e-08	0.00e-00	1.00e-08	0.00e-00	0.00e-00	0.00e-00	1.00e-08	0.00e-00	0.00e-00	0.00e-00
f_3	2.50e-02	5.18e-02	5.17e-02	1.33e-02	1.82e+01	1.21e+02	2.01e-02	1.08e-01	9.95e+03	3.97e+04
f_4	1.00e-08	0.00e-00	1.00e-08	0.00e-00	0.00e-00	0.00e-00	1.00e-08	0.00e-00	1.21e+02	5.34e+02
f_5	1.00e-08	0.00e-00	1.00e-08	0.00e-00	0.00e-00	0.00e-00	1.52e-08	6.96e-09	4.95e-04	2.31e-04
f_6	1.00e-08	0.00e-00	2.88e+01	9.35e-00	0.00e-00	0.00e-00	4.19e+01	7.82e-00	4.34e+01	3.51e-09
f_7	5.47e-01	5.07e-01	3.76e-00	2.59e-00	4.97e-00	5.72e-00	5.44e-01	1.00e-00	1.54e+01	1.37e+01
f_8	2.11e+01	6.04e-02	2.11e+01	6.14e-02	2.11e+01	4.50e-02	2.11e+01	6.29e-02	2.11e+01	3.45e-02
f_9	5.39e-00	2.37e-00	2.47e-00	2.96e-00	7.22e-00	2.29e-00	8.18e-00	3.28e-00	1.76e+01	2.81e-00
f_{10}	1.00e-08	0.00e-00	1.00e-08	0.00e-00	0.00e-00	0.00e-00	1.00e-08	0.00e-00	1.89e-03	3.26e-03
f_{11}	5.63e-00	1.78e-00	5.72e-00	3.27e-00	5.51e-00	2.96e-00	5.94e-00	1.98e-00	6.13e-00	2.29e-00
f_{12}	4.12e-00	7.35e-00	3.28e-00	2.35e-00	5.37e-00	2.54e-00	5.77e-00	1.81e-00	3.33e+01	7.06e-00
f_{13}	7.31e-00	2.58e-00	4.69e-00	3.51e-00	7.60e-00	5.47e-00	5.73e-00	2.64e-00	8.29e+01	1.91e+01
f_{14}	6.07e+02	3.39e+02	4.24e+02	4.59e+02	1.38e+03	5.67e+02	8.59e+02	7.10e+02	5.08e+02	2.19e+02
f_{15}	3.29e+02	3.70e+02	3.90e+02	1.84e+02	1.55e+03	5.48e+02	6.42e+02	2.97e+02	3.32e+03	7.58e+02
f_{16}	3.98e-01	2.73e-01	3.15e-01	2.21e-01	8.78e-01	1.44e-00	6.28e-01	3.53e-01	1.06e-02	3.32e-03
f_{17}	4.04e+01	2.45e+01	2.99e+01	2.53e+01	5.74e+01	2.72e-00	5.75e+01	1.57e-00	6.63e+01	4.76e-00
f_{18}	7.77e+01	9.09e-00	7.29e+01	6.33e+01	1.34e+02	1.00e+02	6.43e+01	8.39e+00	7.83e+01	4.64e-00
f_{19}	5.26e-00	2.37e-01	3.09e-01	1.35e-08	4.56e-00	5.93e-01	3.62e-00	8.79e-01	3.39e-00	3.80e-01
f_{20}	1.01e+01	4.82e-01	4.08e+01	2.83e+01	2.25e+01	1.18e-00	2.44e+01	4.52e-01	1.87e+01	8.18e-01
f_{21}	3.01e+02	6.51e+01	2.36e+02	1.46e+02	1.98e+02	1.40e+01	2.00e+02	0.00e-00	6.95e+02	4.01e+02
f_{22}	9.92e+01	2.71e+01	2.40e+02	2.36e+02	1.45e+03	6.01e+02	5.87e+02	5.62e+02	2.53e+02	1.44e+02
f_{23}	3.73e+02	2.35e+02	6.52e+02	2.84e+02	1.71e+03	8.09e+02	5.57e+02	3.26e+02	3.30e+03	7.24e+02
f_{24}	2.48e+02	7.23e+01	2.03e+02	1.81e+02	2.40e+02	2.04e+01	2.00e+02	5.39e-02	2.21e+02	1.41e+01
f_{25}	2.59e+02	4.98e-00	2.61e+02	6.75e+01	2.48e+02	5.06e-00	2.74e+02	6.24e-00	2.94e+02	8.71e-00
f_{26}	1.76e+02	6.02e+01	2.01e+02	1.14e+02	1.96e+02	1.43e+01	2.41e+02	4.97e+01	2.91e+02	6.28e+01
f_{27}	5.14e+02	1.08e+02	4.10e+02	3.81e+02	7.28e+02	1.44e+02	3.02e+02	1.49e+01	6.39e+02	1.56e+02
f_{28}	4.00e+02	0.00e-00	4.00e+02	0.00e-00	4.00e+02	0.00e-00	4.00e+02	0.00e-00	4.59e+02	4.20e+02

Table 6. Results of the Friedman Test and Holm test.($\alpha = 0.05$)

Friedman test					Holm test			
Algorithm	Rank	χ^2	p value	Diff.?	Algorithm	z	p value	Diff.?
StDE-C	2.50	28.516	0.001	Yes	StDE-C *v.s.* StDE-G	0.64	0.3611	No
StDE-G	2.23							
NBIPOPaCMA	3.21				StDE-C *v.s.* NBIPOPaCMA	1.68	0.0465	Yes
					StDE-G *v.s.* NBIPOPaCMA	2.32	0.0102	Yes
icmaesils	2.95				StDE-C *v.s.* icmaesils	1.06	0.1446	No
					StDE-G *v.s.* icmaesils	1.70	0.0446	Yes
DRMA-LSCh-CMA	4.11				StDE-C *v.s.* DRMA-LSCh-CMA	3.81	0.0002	Yes
					StDE-G *v.s.* DRMA-LSCh-CMA	4.44	0.0001	Yes

6.2.3. Discussions

The simulation results of the StDE-C and the StDE- G on the 28 CEC2013 functions can be summarized as follows.

1. On the unimodal functions f_1–f_5, both the StDE-C and the StDE-G can perform very well, which indicates that both algorithms maintain good performance in terms of their final results.
2. On the basic multimodal functions f_6–f_{20}, when solving the 30-D problems, the StDE-C obtains better results on 4 instances (f_7, f_{11}, f_{15}, f_{20}), and the StDE-G obtains better results on 9 instances (f_8, f_9, f_{12}–f_{14}, f_{16}–f_{19}). When solving the 50-D problems, the StDE-C obtains better results on 6 instances (f_6, f_7, f_{11}, f_{15}, f_{18}, f_{20}), and the StDE-G obtains better results on 7 instances (f_9, f_{12}–f_{14}, f_{16}, f_{17}, f_{19}).
3. On the composition multimodal functions f_{21} − f_{28}, when solving the 30-D problems, the StDE-C obtains better results on 7 instances (f_{21}–f_{23}, f_{25}–f_{28}), while the StDE-G obtains better results on 1 instances f_{24}. When solving the 50-D

problems, the StDE-C obtains better results on 4 instances (f_{23}, f_{25}–f_{27}), while the StDE-G obtains better results on 3 instances (f_{21}, f_{22}, f_{24}).

According to the technical report associated with the CEC2013, the multimodal functions have more local optima than the unimodal functions, and the composition multimodal functions have the most number of local optima. Moreover, the 50-D problems are obviously difficult than the 30-D problems. The simulation results indicate that the StDE-C may outperform StDE-G when the algorithm traps into pseudo optimum, but may not converge as fast as StDE-G when the algorithm does not trap into pseudo optimum. This outcome can be referred as the "no free lunch theorems" for optimization [22], *i.e.*, " any elevated performance over one class of problems is offset by performance over another class". The reason that we get such results is that in StDE-C, the stochastic region of each chromosome is described by Cauchy distribution. And the Cauchy distribution has a higher probability to generate a new position far away from its current center. While in StDE-G, the stochastic region of each chromosome is described by Gaussian distribution. And the Gaussian distribution has a higher probability to generate a new position far away from its current center. While in StDE-G, the stochastic region of each chromosome is described by Gaussian distribution. And the Gaussian distribution has a higher probability to generate a new position close to its current center. So it can be hypothesized that it is better to apply StDE-C when the quality of the final result is more concerned, and to apply StDE-G when the convergence speed is of more importance.

From the above experimental results and discussions, we can observe that the proposed the two StDE algorithms can converge to the optimality region for the utilized test functions. This validates the correctness of *Theorems* 1 and 2. There are inexhaustible optimization problems in the real world, we cannot exhaust all of them to validate the correctness of the proposed theorems, the experiment here however serves as an illustration of usefulness of the theorems and provides a guideline fore researchers to design their algorithms.

7. Conclusions

In this paper, we aimed at designing convergent differential evolution algorithms. In the theoretical aspect, firstly, we proposed and proved that the DE can be modeled using an absorbing Markov sequence. Secondly, we proposed two theorems that give sufficient conditions for DE converging to the optimality region with probability one. From the theorems, we concluded that the first sufficient condition to have DE converge to the optimality region is to make the support set of the whole population covering the entire solution space during the DE execution process. The second is to make the support set of the whole population covers the area that is "closer" to the optimality region during the DE execution process.

In the empirical and application aspect, we proposed a stochastic DE model based on the two proposed theorems. Depending on the properties of the stochastic region, we derived two stochastic DE algorithms, *i.e.*, StDE-C and StDE-G. In StDE-C, the stochastic region is described by Cauchy distribution. The StDE-C satisfies the precondition of the proposed *Theorem* 1 and can be considered as a global search method. In StDE-G, the stochastic region is described by Gaussian distribution. The StDE-G satisfies the precondition of the proposed *Theorem* 2 and can be considered as a local search method. According to the proposed Theorems 1 and 2, both StDE-C and StDE-G can converge to the optimality region. Furthermore both the StDE-C and StDE-G are simple to implement and are similar to the standard DE except for replacing each chromosome's position vector with a pre-described stochastic region. The performance of the StDE-C and StDE-G are tested on the CEC2013 moderate dimensions benchmark functions from three different categories. Simulation results show that both of them are able to find the global solutions for all these test functions. Based on these results, we can

conclude that they can find the optimality region for all the selected test functions. The proposed algorithms exhibit their ability of escaping from pseudo optima and locating global optimality region, thus enhances the usefullness and effectiveness of the proposed theorems.

One main contribution of the paper is the two theorems according to which when the sufficient conditions of either theorem are satisfied, the DE converges to an optimal solution. Two applications are provided to illustrate the easiness and effective- ness of applying the theorems. The future work includes investigating and designing even more effective convergent DE algorithms according to the two theorems, analyzing the convergence speed of DE, and designing algorithms that could speed up the convergence.

Acknowledgement

The authors would like to thank the support of the National Natural Science Foundation of China (61572104, 61103146, 61402076, 61202306, 61572225), Startup Fund for the Doctoral Program of Liaoning Province (20141023), and the Fundamental Research Funds for the Central Universities (DUT15QY26, DUT14QY06).

References

[1] R. Storn, "System design by constraint adaptation and differential evolution", IEEE Transactions on Evolutional Computation, vol. 3, no. 1, **(1999)**, pp. 22-34.
[2] K.V. Price, R. Storn and J. Lampinen, "Differential Evolution: A Practical Approach to Global Optimization", Springer, ISBN 3-540-20950-6, **(2005)**.
[3] M.Z. Ali, N.H. Awad and P.N. Suganthan, "Multi-population differential evolution with balanced ensemble of mutation strategies for large-scale global optimization", Applied Soft Computing, vol. 33, **(2015)**, pp. 304- 327.
[4] M. Weber, F. Neri and V. Tirronen, "A Study on Scale Factor in Distributed Differential Evolution", Information Sciences, vol. 181, no. 12, **(2011)**, pp. 2488-2511.
[5] J. Brest, S. Greiner, B. Boskovic, M. Mernik and V. Zumer, "Self-Adapting Control Parameters in Differ- ential Evolution: A Comparative Study on Numerial Benchmark Problems", IEEE Transactions on Evolu- tionary Computation, vol. 10, no. 6, **(2006)**, pp. 646-657.
[6] A.K. Qin, V.L. Huang and P.N. Suganthan, "Differential Evolution Algorithm with Strategy Adaptation for Global Numerical Optimization", IEEE Transactions on Evolutionary Computation, vol. 13, no. 2, **(2009)**, pp. 398-417.
[7] S. Hui and P.N. Suganthan, "Ensemble and arithmetic recombination-based special differential evolution for multimodal optimization", IEEE Transactions on Cybernetics, vol. 46, no. 1, **(2016)**, pp. 64-74.
[8] G.H. Wu, R. Mallipeddi and P.N. Suganthan, "differential evolution with multi-population based ensemble of mutation strategies", Infomration Sciences, vol. 329, no. SI, **(2016)**, pp. 329-345.
[9] F. Neri and E. Mininno, "Memetic compact differential evolution for Cartesian robot control", IEEE Computational Intelligence Magazine, vol. 5, no. 2, **(2010)**, pp. 54-65.
[10] D. Zaharie, "Statistical properties of differential evolution and related random search algorithms", Proceedings of the International Conference on Computational Statistics, Porto, Portugal, **(2008)**, pp.473-485.
[11] D. Zaharie, "Influence of crossover on the behavior of differential evolution algorithms", Applied Soft Computing, vol. 9, no. 3, **(2009)**, pp 1126-1138.
[12] 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, **(2009)**, pp. 1-20.
[13] K. Zielinski, D. Peters and R. Laur, "Run time analysis regarding stopping criteria for differential evolution and particle swarm optimization", Proceedings of the 1st international conference on Process/System Mod- elling/Simulation/Optimization, Athens, Greece, **(2005)**, pp. 235-243.
[14] F. Xue, A.C. Sanderson and R.J. Graves, "Modeling and convergence analysis of a continuous multiobjective differential evolution algorithm", Proceedings of the 2005 IEEE Congress on Evolutionary Computation, Edinburgh, Scotland, **(2005)**, pp. 228-235.
[15] F. Solis and R. Wets, "minimization by random search techniques, mathematics of Operations Research", vol. 6, no. 1, **(1981)**, pp. 19-30.
[16] J. Liang, B. Qu, P.N. Suganthan and Alfredo G. Hernandez-Diza, "Problem definitions and evaluation criteria for the CEC 2013 special session on real parameter optimization", Technical Report, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou, China, **(2012)**, pp. 1-13.

[17] H. Ge, L. Sun, X. Yang, S. Yoshida and Y. Liang, "Cooperative differential evolution with fast variable interdependence learning and cross-cluster mutation", Applied Soft Computing, vol. 36, no. 1, **(2015)**, pp. 300-314.
[18] I. Loshchilov, "CMA-ES with restarts for solving CEC 2013 benchmark problems", Proceedings of the 2013 IEEE Congress on Evolutionary Computation, Cancun, Mexico, **(2013)**, pp. 369-376.
[19] T. Liao and T. Stutzle, "Benchmark results for a simple hybrid algorithm on the CEC 2013 benchmark set for real-parameter optimization", Proceedings of the 2013 IEEE Congress on Evolutionary Computation, Cancun, Mexico, **(2013)**, pp. 1938-1944.
[20] B. Lacroix, D. Molina and F. Herrera, "Dynamically up- dated region based memetic algorithm for the 2013 CEC special session and competition on real parameter single objective optimization", Proceedings of the 2013 IEEE Congress on Evolutionary Computation, Cancun, Mexico, **(2013)**, pp. 1945-1951.
[21] S. Garcia, D. Molina, M. Lozano and F. Herrera, "A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behavior: a case study on the CEC 2005 special session on real parameter op- timization", Journal of Heuristics, vol. 15, no. 6, **(2009)**, pp. 617- 644.

