

# A Comprehensive Analysis on Optimization Performance of Chaotic Evolution and Its Parameter Distribution

Pei, Yan  
University of Aizu

Takagi, Hideyuki  
Faculty of Design, Kyushu University

Zhao, Qiangfu  
University of Aizu

Liu, Yong  
University of Aizu

<https://hdl.handle.net/2324/1905851>

---

出版情報 : Proceedings 2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp.3481-3486, 2014-10-08. IEEE

バージョン :

権利関係 :

# A Comprehensive Analysis on Optimization Performance of Chaotic Evolution and Its Parameter Distribution

Yan Pei\*, Hideyuki Takagi<sup>†</sup>, Qiangfu Zhao\* and Yong Liu\*

\*School of Computer Science and Engineering, The University of Aizu, Aizu-wakamatsu, Japan 965-8580

Email: peiyan@u-aizu.ac.jp, qf-zhao@u-aizu.ac.jp, yliu@u-aizu.ac.jp

<sup>†</sup>Faculty of Design, Kyushu University, Fukuoka, Japan 815-8540

Email: takagi.hideyuki.457@m.kyushu-u.ac.jp

**Abstract**—In this paper, we analyse and discuss the relationship between optimization performance of chaotic evolution (CE) algorithm and distribution characteristic of chaotic parameter. CE is an evolutionary computation algorithm that simulates chaotic motion of a chaotic system in a search space for implementing optimization. However, its optimization performance, internal process mechanism and optimization principle are not well studied. In this paper, we investigate distribution characteristics of chaotic systems, which support chaotic parameter in CE algorithm. Compared with other two parameter generators, i.e., a quadratic-like random generator and a uniform random generator, CE algorithm with chaotic parameter generated by the logistic map ( $\mu = 4$ ) shows better optimization performance significantly. We also make an algorithm comparison with differential evolution and an algorithm ranking by Friedman test and Bonferroni-Dunn test. The related topics on relationship between optimization performance of CE algorithm and chaotic parameter distribution are analysed and discussed. From these analyses and discussions, it indicates that chaotic parameter distribution is a significant factor that influences optimization performance of CE algorithm.

**Keywords**—Chaotic Evolution, Chaotic Optimization, Chaos Theory, Natural Computing, Evolutionary Computation

## I. INTRODUCTION

Evolutionary computation (EC) is a set of algorithms that are inspired from biological, physical and mathematical phenomena and mechanisms. Roughly speaking, there are two research directions involved in EC community. One is to extend application scale of EC algorithms, and the other is to improve their optimization capability. They are considered as application and theoretical researches of EC algorithm [4].

In the subjects of theoretical research aspect on EC algorithm, three directions for current research received special focus [3]. First is approximating fitness landscape, the objective here is to find out promising search region and structure of optimized problem so as to search for a global optimum easily [5], [6]. Second is developing new mechanisms for enhancing optimization capability of well designed EC algorithms [7]. Third is to create a new EC algorithm by inspirations from biological, physical, mathematical or other phenomena and mechanisms [2].

When we try to explain and understand our natural world and principle, there are three methodologies and philosophies

from our best knowledge and capability. They are determinism, probability and chaos. In optimization field, deterministic and stochastic optimization algorithms are well support by fundamentals of determinism and probability, respectively. However, chaotic optimization algorithm is not well studied in chaotic field. Chaotic evolution can be considered as one of its breakthrough proposal in optimization field that are supported by chaotic philosophy and methodology. It simulates chaotic motion in a search space to implement exploration and exploitation function of EC algorithm. In CE optimization framework, there is a parameter, *Chaotic Parameter*, from a chaotic system that implements actual search function. A randomly generated direction factor guides search direction and implements a chaotic search of CE. However, the relationship between chaotic parameter's distribution and optimization performance of CE is not well studied. The internal optimization mechanism of CE is not clear as well. In this paper, we try to investigate these research subjects mentioned above and discuss related issues.

The structure of this paper after this introductory section is as follows. Section II introduces some background knowledge about differential evolution (DE) and chaotic evolution (CE) algorithms, and makes comprehensive discussions on their differences. Analyses and discussions on the subjects of relationship between distribution characteristics of CE and their performance are conducted in sections III and IV. We introduce two chaotic parameter generators of CE to compare with CE and DE and make an algorithm ranking by Friedman test and Bonferroni-Dunn test. Finally, we conclude the whole work and present future opportunities, open topics and future works in section V.

## II. A BRIEF OVERVIEW ON DIFFERENTIAL EVOLUTION AND CHAOTIC EVOLUTION

Here, we briefly introduce the algorithm frameworks of DE and CE. Most of the contents are adapted from [8] for DE and [2] for CE. Finally, we make a comprehensive remark on the difference between DE and CE.

### A. Differential Evolution

DE is a population-based optimization algorithm [8], which controls the balance of global exploration and local exploitation automatically due to differential vector. It searches for a

global optimum using differential vector from two individuals. Average length of differential vector is in proportion to the distribution size of individuals. Each one parent individual generates one offspring and the offspring replaces its parent when its fitness is better than that of the parent.

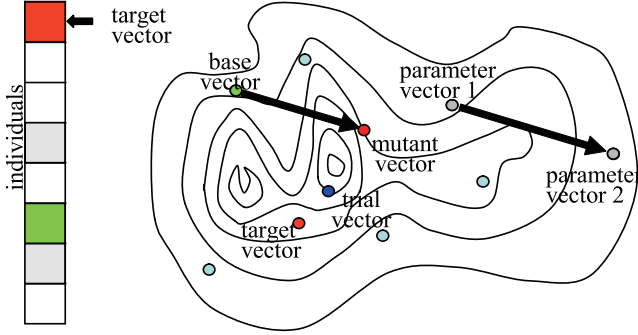


Fig. 1. Differential evolution algorithm.

Suppose that an array on the left side of Fig. 1 means individuals, contour line at the right side is a fitness landscape, and circles on the landscape are individuals. The following description outlines primary steps of a DE implementation. It repeats until a satisfied solution is found or generation number reaches to maximum.

- (1) Choose one individual to be a target vector ( $x_{i,G}$ ).
- (2) Select two other individuals ( $x_{r_2,G}$  and  $x_{r_3,G}$ ) as parameter vectors randomly and derive a differential vector from them.
- (3) Select the best individual from the rest of individuals or another individual randomly as a base vector ( $x_{r_1,G}$ ).
- (4) Create a mutant vector ( $v_{i,G}$ ) by adding a weighted differential vector to the base vector.
- (5) Generate a trial vector by crossing the target vector and the mutant vector.
- (6) Compare the fitness of the target vector and the trial vector, and select whichever one is better as the offspring in the next generation.
- (7) Go to step (1) and generate other offspring until all individuals are processed using the same operations. Then proceed with the next generation.

$$v_{i,G} = x_{r_1,G} + F * (x_{r_2,G} - x_{r_3,G}) \quad (1)$$

$G$  means generation, and  $i, r_1, r_2, r_3$  are indexes of individuals, and  $i \neq r_1 \neq r_2 \neq r_3$ . The steps (1) – (4) are summarized as Eq. (1), where  $F$  is scale factor. This description demonstrates that the DE algorithm is easily implemented.

### B. Chaotic Evolution

A chaotic ergodicity based EC algorithm, *chaotic evolution*, was proposed and studied recently [1], [2]. It simulates chaotic motion in a search space for optimization by inspired from chaotic motion of non-linear system and chaotic optimization implementation in chaos field. Because the chaotic motion has an ergodic property, the proposed algorithm can guarantee its global convergence partially. The scalability of algorithm

**Algorithm 1** Chaotic Evolution Algorithm.  $PS$ : population size;  $Dim$ : dimension;  $D$ : direction factor;  $DR$ : direction factor rate;  $CP$ : chaotic parameter;  $G$ : generation;  $maxIter$ : maximum generation;  $i$ : index of individual;  $j$ : index of dimension.

```

1: Generate an initial population.
2: Evaluate the fitness for each individual.
3: /*D and CP initialization*/
4: for i = 1 to PS do
5:   for j = 1 to Dim do
6:     /*DR as a random value*/
7:     if rand[0, 1) < DR then
8:        $D_{i,j} = -1$ 
9:     else
10:       $D_{i,j} = +1$ 
11:    end if
12:     $CP_{i,j} = \text{rand}(0,1)$ 
13:  end for
14: end for
15: /*Chaotic Search*/
16: for G = 1 to maxIter do
17:   for i = 1 to PS do
18:      $k = \text{rand}(1, Dim)$ 
19:     for j = 1 to Dim do
20:       if rand[0, 1) <  $C_r$  or  $j == k$  then
21:          $mutant_{i,j} = target_{i,j} * (1 + D_{i,j} * CP_{i,j})$ 
22:          $chaotic_{i,j} = mutant_{i,j}$ 
23:       else
24:          $chaotic_{i,j} = target_{i,j}$ 
25:       end if
26:     end for
27:   /*Selection*/
28:   for i = 1 to PS do
29:     if  $f(chaotic_i) < f(target_i)$  then
30:       replace  $target_i$  with  $chaotic_i$ 
31:     end if
32:   end for
33: end for
34: /*D and CP update*/
35: for i = 1 to PS do
36:   for j = 1 to Dim do
37:      $CP_{i,j} = \text{ChaoticSystem}(CP_{i,j})$ 
38:     if rand[0, 1) < DR then
39:        $D_{i,j} = -1$ 
40:     else
41:        $D_{i,j} = +1$ 
42:     end if
43:   end for
44: end for
45: end for
46: return the optimum

```

is better than other EC algorithms by introducing different chaotic systems into its framework.

$$x_n = \mu * x_{n-1} * (1 - x_{n-1}) \quad (2)$$

TABLE I. 12 BENCHMARK FUNCTIONS USED IN EXPERIMENTAL EVALUATIONS (UNI=UNI-MODAL, MUL=MULTI-MODAL, GB=GLOBAL ON BOUNDS, NS=NON-SEPARABLE, AND S=SEPARABLE). THE RELATED PARAMETERS OF BENCHMARK FUNCTIONS ARE DESCRIBED IN [9].

No.	Name	Form	Range	Optimum(base value)	Characters
F1	Sphere	$f(x) = \sum_{i=1}^n x_i^2 + base$	[-100,100]	-450	Uni-S
F2	Schweffel 1.2	$f(x) = \sum_{i=1}^n (\sum_{j=1}^i (x_j))^2 + base$	[-100,100]	-450	Uni-NS
F3	Elliptic	$f(x) = \sum_{i=1}^n (10^6)^{\frac{i-1}{n-1}} x_i^2 + base$	[-100,100]	-450	Uni-NS
F4	F2 with Noise	$f(x) = \sum_{i=1}^n (\sum_{j=1}^i (x_j))^2 * (1 + 0.4 N(0, 1) ) + base$	[-100,100]	-450	Uni-NS
F5	Schweffel 2.6 GB	$f(x) = Max A_i x - B_i  + base$	[-100,100]	-310	Uni-NS
F6	Rosenbrock	$f(x) = 100(x_1^2 - x_2)^2 + (1 - x_1)^2 + base$	[-100,100]	390	Mul-NS
F7	Griewank	$f(x) = 1 + \sum_{i=1}^n \frac{x_i^2}{4000} - \prod_{i=1}^n \cos(\frac{x_i}{\sqrt{i}}) + base$	[0,600]	-180	Mul-NS
F8	Ackley GB	$f(x) = -20 \exp(-0.2(\frac{1}{n} \sum_{i=1}^n x_i^2)^{\frac{1}{2}}) - \exp(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)) + 20 + e + base$	[-32,32]	-140	Mul-NS
F9	Rastrigin	$f(x) = (10n) + \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i)) + base$	[-5,5]	-330	Mul-S
F10	Weierstrass	$f(x) = \sum_{i=1}^n (\sum_{k=0}^{k=i} [a^k \cos(2\pi b^k (x_i + 0.5))]) - n \sum_{k=0}^{k=i} [a^k \cos(2\pi b^k * 0.5)] + base$	[-0.5,0.5]	90	Mul-NS
F11	Schweffel 2.13	$f(x) = \sum_{i=1}^n (\sum_{j=1}^i (a_{ij} \sin x_i + b_{ij} \cos x_i))^2 + base$	[-100, 100]	-460	Mul-NS
F12	Expanded F8F2	$f(x) = F8F2(x_1, x_2, \dots, x_n) = F8(F2(x_1, x_2) + \dots + F8(F2(x_n, x_1)) + base$	[-3,1]	-130	Mul-NS

Algorithm 1 presents primary steps of a CE implementation [2], which the following description outlines. There are population size ( $PS$ ), chaotic parameter ( $CP_i$ ), search direction ( $D_i$ ) and crossover rate ( $C_r$ ) as parameters in the CE algorithm framework that need to be set. If a non-linear system can result in chaos, it can be used for generating  $CP_i$ . So there are a variety of CE implementations. In this paper, we use the logistic map (Eq. (2),  $x_0 \in (0, 1)$ ,  $\mu = 4$  and  $\mu \neq 0, 0.25, 0.50, 0.75, 1$ ) as  $CP_i$  generator. A brief explanation of algorithm 1 is summarized as follow.

- (1) Line 1 is for generating initial population and line 2 is for evaluating each individual.
- (2) From lines 4 to 14, chaotic parameter ( $CP_i$ ) and search direction ( $D_i$ ) are initialized with  $rand(0, 1]$ , and  $\{+1, -1\}$  due to a direction factor rate, respectively.
- (3) It is the main search implementation of a CE algorithm from lines 16 to 33. Line 21 shows the mutant vector production. From lines 19 to 26, it is the crossover and mutation operations of CE. It is the selection operation from line 28 to 32, and only a new generated chaotic vector needs a fitness evaluation in line 29.
- (4) Lines 35 to 44 show the process of updating parameters  $CP_i$  and  $D_i$  in each generation.
- (5) Line 46 returns the optimum.

### C. Difference between DE and CE

There are many differences between DE and CE. First of all, the design philosophies of DE and CE come from different inspirations. DE takes account of differential vector, where fitness landscape is hidden behind differential vector, in a search space; and use it to implement exploration and exploitation. CE is an algorithm that simulates chaotic ergodic motion in a search space to implement search function. So optimization capability of CE depends on statistical characteristics of chaotic system used in CE algorithm framework. More critical issue is DE is a stochastic optimization algorithm whose fundamental is probability theory. However, CE is a chaotic optimization algorithm which is supported by philosophy and methodology of chaos theory. Second, the differences of DE and CE are their concrete implementations. Eq. (1) presents the main search function of DE, where the mutant vector is decided by differential vector. However, step 21 in Algorithm 1 sketches production process of a chaotic vector, which is made from perturbation by chaotic parameter from a

chaotic system. Third, from the number of function call, the total number of fitness evaluations in CE is the same as that in DE [2]. CE algorithm calls a fitness function at two places. One is after population initialization, and all individuals need to be evaluated by a fitness function. The other is when the chaotic vector is compared to the target vector, the chaotic vector needs to be evaluated by a fitness function (Algorithm 1). Forth, they are quite different from the viewpoint of algorithm taxonomy. DE use differential vector that uses of population information or landscape information to implement its search mechanism, however, CE searches for a global optimum by using chaotic perturbation independently. DE is an algorithm of EC, but CE can be categorized as an algorithm of partial swarm optimization, if it does not use any population information or fitness landscape. In a limit condition, the minimum population size of DE is four and that of CE is only one.

TABLE II. DISTRIBUTION CHARACTERISTICS OF A LOGISTIC MAP ( $\mu = 4$ ), A QUADRATIC-LIKE DISTRIBUTION ( $N(0, 0.15^2) + N(1, 0.15^2)$ ) AND AN UNIFORM DISTRIBUTION.

Interval	logistic	quadratic-like	uniform
(0, 0.1]	18.60%	22.80%	11.20%
(0.1, 0.2]	7.50%	16.80%	9.70%
(0.2, 0.3]	8.60%	6.10%	9.60%
(0.3, 0.4]	7.70%	2.30%	9.30%
(0.4, 0.5]	5.10%	0.40%	11.30%
(0.5, 0.6]	6.20%	0.80%	11.80%
(0.6, 0.7]	7.00%	1.60%	9.80%
(0.7, 0.8]	8.90%	6.70%	9.10%
(0.8, 0.9]	9.70%	15.90%	9.40%
(0.9, 1]	20.70%	26.60%	8.80%

## III. EXPERIMENTAL EVALUATIONS

### A. Remaining Problem and Experimental Design

Although CE has shown a competitive optimization performance from [2], its search principle and internal optimization mechanism are still not well studied. In this paper, we conduct some experiential evaluations to investigate the relationship between optimization performance of CE and distribution characteristic of chaotic parameter.

Two additional generators of chaotic parameter are introduced in our competitive evaluations. First is a quadratic-like distribution that is implemented by two normal distributions ( $N(0, 0.15^2) + N(1, 0.15^2)$ ) in the interval of  $[0, 1]$ . Eq. (3) is its probability mass function and  $N(\mu, \sigma^2)$  is shown in Eq. (4), where  $\mu$  is mean and  $\sigma^2$  is standard deviation. Second is an uniform distribution whose probability mass function is shown in Eq. (5) and  $a = 0$ ,  $b = 1$ . Besides these two generators for

implementing two algorithms, we also compare CE with DE and the two algorithms. The characteristics of quadratic-like distribution, uniform distribution and the logistic map output distribution ( $\mu = 4$ ) is listed in Table. II.

$$f(x) = N(0, 0.15^2) + N(1, 0.15^2) \quad (3)$$

$$N(\mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} \exp -\frac{(x - \mu)^2}{2\sigma^2} \quad (4)$$

$$f_{I[a,b]}(x) = \frac{1}{b-a} I[a,b] \quad (5)$$

TABLE III. DE AND CE EXPERIMENT PARAMETERS SETTING.

population size	50
max. search generation	$10^9$
dimensions of benchmark functions, $D$	10
# of trial runs	30
DE	DE/best/1/bin
scale factor $F'$	0.3
crossover rate	0.7
CE direction factor rate $DR$	random
CE crossover rate	1.0

### B. Experimental Setting and Results

We evaluate these algorithms using 12 benchmark functions. The definition of benchmark functions, ranges, the global optimum and characteristics are listed in Table I. Some of them are adopted from [9]. We test each benchmark function for 100 generations with 30 trial runs. The parameter setting of CE and DE is in Table III. Fig. 3 shows average convergence curves of best fitness values from 30 trial runs for all 12 benchmark functions. Table V shows their means. The marks † and ‡ mean CE-logistic method significantly outperforms CE-quadratic and CE-uniform, respectively, by Wilcoxon signed test ( $p < 0.05$ ). The marks  $\Delta$  and  $\square$  mean CE-quadratic and CE-uniform outperform DE, respectively, by Wilcoxon signed test ( $p < 0.05$ ). Abbreviations used in Figures 3 and Table V are given in Table IV.

TABLE IV. ABBREVIATIONS USED IN EXPERIMENTAL EVALUATIONS.

abbreviations	Meaning
DE	standard DE
CE-logistic	chaotic evolution with $CP$ updated by Eq. (2).
CE-quadratic	chaotic evolution with $CP$ updated by Eq. (3).
CE-uniform	chaotic evolution with $CP$ updated by Eq. (5).

## IV. ANALYSES AND DISCUSSIONS

### A. Comparison of CE and Its Competitors

The output distribution of the logistic map has a characteristic that most of the outputs cover the intervals  $[0, 0.1]$  and  $[0.9, 1]$  (Table II). From exploration and exploitation viewpoints, this characteristic decides the CE's exploration and exploitation functions. To confirm this hypothesis, we compare the optimization performance by CE with chaotic parameter  $CP$  from a logistic map ( $\mu = 4$ ), a quadratic-like distribution ( $N(0, 0.15^2) + N(1, 0.15^2)$ ) and an uniform distribution. We abbreviate the CE algorithms with  $CP$  set by these distribution in Table IV.

The experiment uses 10-D functions from Table I with 50 individuals, 100 generations and 30 running trails. Because most of the benchmark functions obtain the global optimum before the 100th generation, we conduct Wilcoxon signed rank test at the 10th generation. Table V shows means of 12 benchmark functions and results of Wilcoxon signed rank test ( $p < 0.05$ ). Most of the CE-logistic method significantly outperforms CE-uniform, and significant difference could not found between CE-logistic algorithm and CE-quadratic algorithm. It indicates that output distribution characteristic of the logistic map decides optimization performance of CE algorithm. CE-quadratic and CE-uniform outperform DE in some benchmark tasks, it indicates that evolution strategy-like algorithm (CE-quadratic and CE-uniform) can obtain better optimization performance.

TABLE V. MEANS OF CE ALGORITHM WITH  $CP$  GENERATED BY A LOGISTIC MAP ( $\mu = 4$ ) (CE-LOGISTIC), A QUADRATIC-LIKE DISTRIBUTION ( $N(0, 0.15^2) + N(1, 0.15^2)$ ) (CE-QUADRATIC), AN UNIFORM DISTRIBUTION (CE-UNIFORM) AND DE.

Function	CE-logistic	CE-quadratic	CE-uniform	DE
F1	-342.18 †	-134.26	-74.69	-127.17
F2	20.41 ‡	340.60 $\Delta$	547.68 $\square$	934.95
F3	234550.49 ‡	446507.86 $\Delta$	1485224.95	2223488.16
F4	-13.31 †	107.20 $\Delta$	299.30 $\square$	1288.93
F5	335.98 ‡	865.73 $\Delta$	1146.03 $\square$	1641.78
F6	1950824.34	2589855.58 $\Delta$	4794977.98	3731426.17
F7	-178.12 †	-176.27 $\Delta$	-175.62	-176.09
F8	-135.68 †	-134.63 $\Delta$	-132.87	-133.00
F9	-320.12 †	-316.09 $\Delta$	-306.51 $\square$	-280.64
F10	92.09 ‡	92.94 $\Delta$	93.81 $\square$	94.90
F11	85301.58	91033.36	88215.97	89324.51
F12	-124.42 †	-124.61	-123.47	-124.01

### B. Algorithms Ranking by Friedman Test

We apply Friedman test and a Bonferroni-Dunn test on CE-logistic, CE-quadratic, CE-uniform and DE to make an algorithm ranking. Table VI shown the results of Friedman test by measure of mean ranking. The CE-logistic method and CE-quadratic method are almost with the same ranking, and better than CE-uniform and DE. Except F11, all  $p$  values from Friedman test are less than 0.05, that means there are significant difference among these algorithms. To check the significant level of all the algorithms, we apply an additional Bonferroni-Dunn test to calculate the critical difference ( $CD$  in Eq. (6)) for comparing their differences in significant level of  $\alpha < 0.05$ .

$$CD = q * \sqrt{\frac{k * (k + 1)}{6 * N}} \quad (6)$$

In Eq. (6), parameters  $k$  and  $N$  are the number of algorithm and number of benchmark task, respectively. They are  $k = 4$  and  $N = 11$  in our experimental evaluations (Because  $p$  value of F11 from Friedman test is more than 0.05, the average of mean ranking only calculates F1-F10 and F12, i.e. without F11. We use these mean ranking value in the Bonferroni-Dunn test in Fig.2. ). When  $\alpha < 0.001$ ,  $\alpha < 0.01$ , and  $\alpha < 0.05$ ,  $q$  is 3.765, 3.144 and 2.639, respectively, from Table B15 of [10]. Fig. 2 sketches the results of Bonferroni-Dunn test. From the result of Fig. 2, there is a significant difference between these algorithms in the significant levels of  $\alpha < 0.001$ . We can conclude that CE algorithms with  $CP$  generated

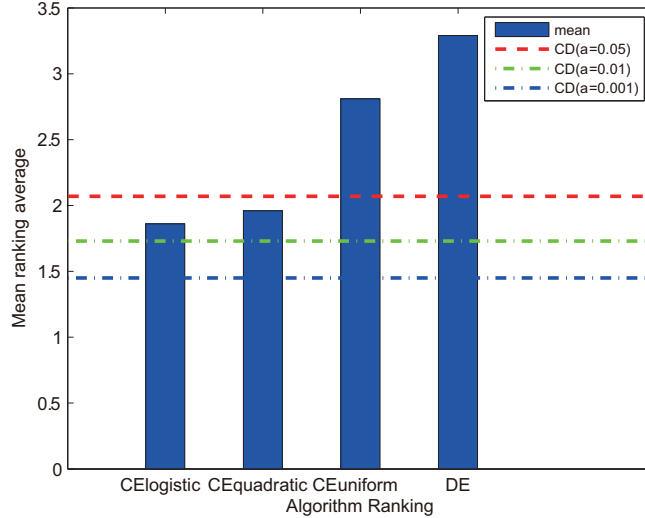


Fig. 2. Rankings obtained through the Friedman test and graphical representation of the Bonferroni-Dunn's procedure by taking CE-logistic as a control method.

by the logistic map ( $\mu = 4$ ) (CE-logistic) and a quadratic-like distribution ( $N(0, 0.15^2) + N(1, 0.15^2)$ ) (CE-quadratic) have the same optimization performance. This conclusion indicates distribution characteristic of  $CP$  generator decides optimization performance of CE.

TABLE VI. MEAN RANKING OF CE ALGORITHM WITH  $CP$  GENERATED BY A LOGISTIC MAP ( $\mu = 4$ ) (CE-LOGISTIC), A QUADRATIC DISTRIBUTION ( $N(0, 0.15^2) + N(1, 0.15^2)$ ) (CE-QUADRATIC), AN UNIFORM DISTRIBUTION (CE-UNIFORM) AND DE BY FRIEDMAN TEST ( $p < 0.05$ ).

Function	CE-logistic	CE-quadratic	CE-uniform	DE
F1	1.87	2.17	2.83	3.13
F2	1.97	2.10	2.67	3.27
F3	1.57	1.80	3.10	3.53
F4	1.97	1.50	2.77	3.77
F5	1.67	2.07	2.77	3.50
F6	2.30	1.80	3.00	2.90
F7	1.83	2.00	2.90	3.27
F8	1.93	2.13	2.90	3.03
F9	1.83	1.80	2.57	3.80
F10	1.70	2.13	2.77	3.40
F11	2.43	2.22	2.58	2.77
F12	1.87	2.17	2.83	3.13
Ave. without F11	1.86	1.96	2.82	3.33

## V. CONCLUSION AND FUTURE WORKS

In this paper, we have investigated one topic on the relationship between distribution characteristic of chaotic parameter and their optimization performance. By introducing two generators of chaotic parameter, we compare DE with CE implemented by these two  $CP$  generators and the logistic map. From experimental evaluation, we can conclude that distribution characteristic of chaotic parameter is a significant factor that can influence the optimization capability of CE algorithm. We also discuss the difference between CE and DE.

CE is a new EC algorithm, and its optimization mechanism behind current optimization results are not clear. In the CE optimization framework, there are several parameters that need to be set. However, there is "no free lunch" that a certain combination of the parameters can be used in any real world

problems or benchmark tasks. We should make the relationship between these parameters and optimization performance clear for designing a more efficient CE algorithm. We will continue to conduct related research topics on this issue.

## REFERENCES

- [1] Pei Yan, "A chaotic ergodicity based evolutionary computation algorithm", The 2013 9th Int. Conf. on Natural Computation, pp.454–459, Shenyang, China (2013).
- [2] Pei Yan, "Chaotic Evolution: fusion of chaotic ergodicity and evolutionary iteration for optimization", Natural Computing, Springer, vol.13(1), pp.79–96 (2014).
- [3] Pei Yan, "Study on efficient search in evolutionary computation", Ph.D Dissertation, Kyushu University, Fukuoka, Japan (2014).
- [4] Pei Y., Takagi H., "A survey on accelerating evolutionary computation approaches", 2011 Int. Conf. on Soft Computing and Pattern Recognition, pp.201–206 (2011).
- [5] Pei Y, Takagi H., "Fourier analysis of the fitness landscape for evolutionary search acceleration", 2012 IEEE Congress on Evolutionary Computation, pp 1–7 (2012).
- [6] Pei Y, Takagi H., "Accelerating iec and ec searches with elite obtained by dimensionality reduction in regression spaces", Evolutionary Intelligence vol.6(1):27–40 (2013).
- [7] Pei Y, Takagi H., "Triple and quadruple comparison-based interactive differential evolution and differential evolution", The twelfth workshop on Foundations of genetic algorithms XII, pp.173–182 (2013).
- [8] Storn R., and Price K., "Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces", Journal of Global Optimization, vol.11(4), pp.341–359 (1997).
- [9] Suganthan P., Hansen N., Liang J., Deb K., Chen Y., Auger A. and Tiwari S., "Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization", Technical Report, Nanyang Technological Uni., Singapore (2005).
- [10] Zar Jerrold H., "Biostatistical analysis (5th Edition)", Pearson Prentice-Hall, Upper Saddle River, NJ, USA (2007).

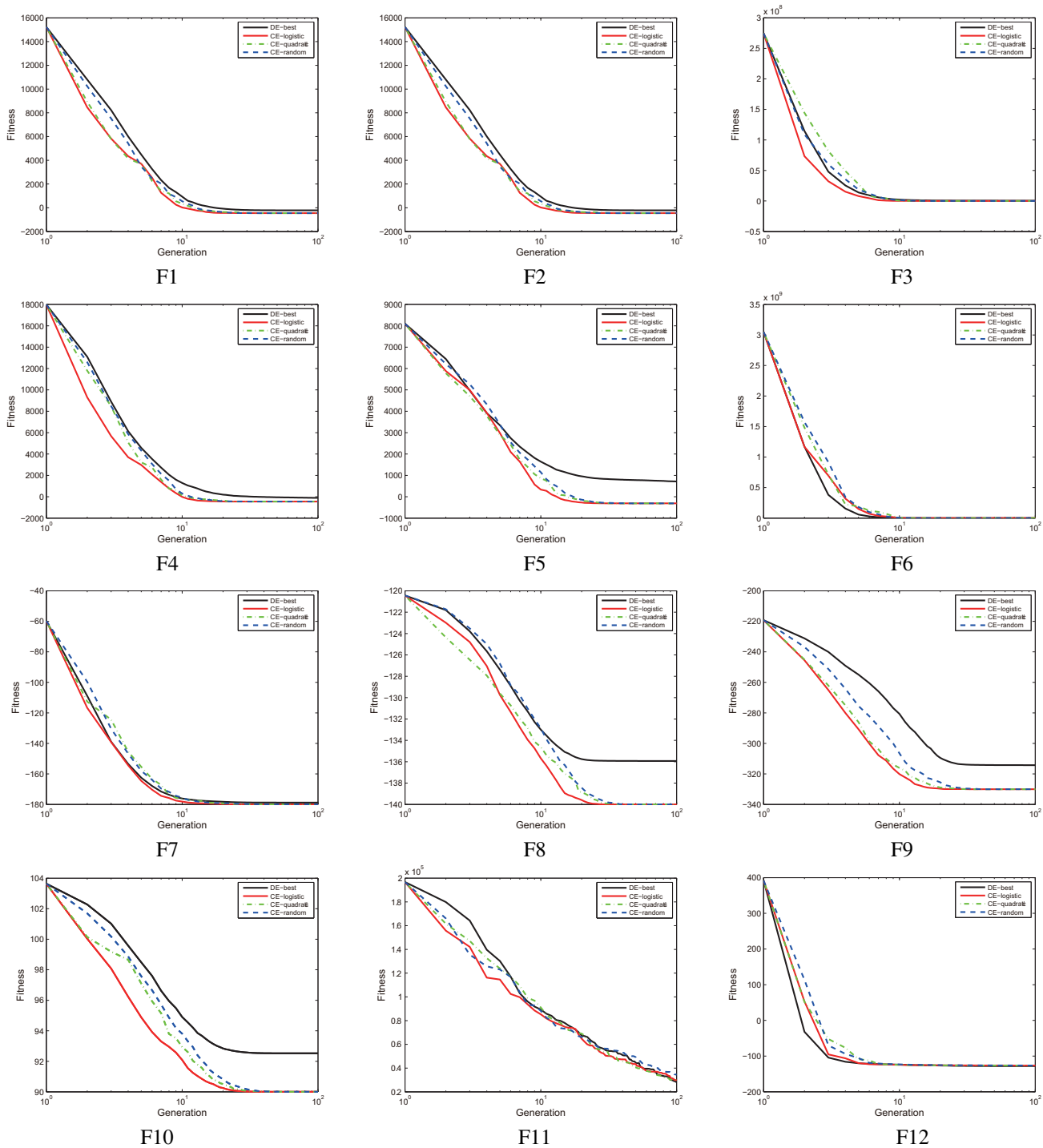


Fig. 3. Average convergence curves of 30 trial runs for F1-F12.