# Principal Coordinate Strategy: A Novel Adaptive Control Strategy for Differential Evolution

Yifeng Gao School of Science Xidian University gaoyfeng2010@hotmail.com

# ABSTRACT

Differential evolution (DE) algorithm has a wide use in optimization problems, whose performance is closely related to the separability of the fitness function. In this paper, we propose Principle Coordinate (PC) strategy, a new adaptive control strategy to improve DE's performance. PC attempts to maximize the fitness function's separability and make crossover operator more robust through coordinate rotation. In PC, Principal Component Analysis (PCA) is adopted to draw the ideal coordinate system from the difference vectors distribution. In the numerical experiments, PC is combined with two versions of classical DE algorithms to test its ability. The first experiment measures the accuracy of the coordinate system obtained by PC. In the second experiment, four benchmark functions and an engineering project are used to evaluate PC's efficiency. The results show that PC improves DE's efficiency, robustness and stability.

### **Categories and Subject Descriptors**

G.1.6 [Numerical Analysis]: Optimization

## **General Terms**

Evolutionary Algorithm

#### Keywords

Differential Evolution, Coordinate System, Separability

## 1. INTRODUCTION

Differential evolution (DE) algorithm is a heuristic algorithm, which is designed to tackle complex optimization and searching problems by iteratively improving a batch of candidate solutions. DE is firstly introduced in a technical report by K. Price and R. Storn in 1995 [18], and two years later it is described in details in a journal paper [19]. Since then, DE has been attracting much attention of researchers, which results in a rapid development of DE.

GECCO'12 Companion, July 7–11, 2012, Philadelphia, PA, USA.

Copyright 2012 ACM 978-1-4503-1178-6/12/07 ...\$10.00.

Dan Lv School of Telecommunications Engineering Xidian University danlv19901005@gmail.com

Although it is widely believed that DE has many advantages over most other evolutionary algorithms [5], classical DE still has some drawbacks. Under some fitness function landscapes, DE is likely to be deceived by local optima [11]. Besides, DE lacks sufficient robustness with strong interdependent variables [20]. The work [11] also indicates that DE has a limited ability to move its solutions over long distance when its population is clustered in a relatively small part of searching space.

Some researchers have already made their efforts to remove classical DE's weaknesses. The works [3, 22] show that DE can be combined with Niche technology to design adaptive penalty function, which can be automatically adjusted by factors such as population, diversity of individuals. In this way, the premature convergence phenomenon is avoided. Some variants of DE develop adaptive control parameters in DE's operators to improve DE's performance, such as jDE [4], which has adaptive mutation parameter Fand crossover rate Cr. Some works improve the original DE operators to more robust and flexible ones. For example, trigonometric mutation operator in [7] and self-adaptive different evolution (SaDE) in [13]. DE has also been hybridized with some other optimization algorithms, such as ant optimization algorithm (AOA) [6], simulated annealing (SA) [10], artificial neural network (ANN) [9]. The hybrid algorithms yield better performance than classical DE.

Besides the forgoing works, some researches reveal that DE's performance is related to the coordinate system [2, 15, 16], because the separability of the fitness function is affected by the coding system, which is uniquely defined by the coordinate system. Separable means that a function can be divided into several sub-functions, each of which only relates to single variable. Some previous works [17, 14] indicate that DEs are more successful with separable fitness functions than non-separable ones because crossover operator becomes more efficient with separable fitness functions. Therefore, an appropriate coordinate system can improve the separability of fitness function. In this way, DE can yield a better performance.

In this paper, a novel control strategy, Principal Coordinate (PC) strategy is proposed, which improves DE's performance by adaptively adjusting the coordinate system into the most suitable one for the fitness function. In this strategy, principal component analysis [1] (PCA) is adopted to obtain the best coordinate system from the current difference vectors distribution. In the experiments, the hybridized algorithms of classical DEs and the PC strategy are used to test PC's accuracy as well as efficiency. Two benchmark

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

functions with the known ideal coordinate systems prove PC has good ability to track suitable coordinate system for the fitness function. Then four benchmark functions and an engineering project are chosen to draw a comparison between the hybridized algorithms and classical DEs, in which the hybridized algorithms outperform the classical DEs. The experiment results show PC is a promising control strategy.

This paper is organized as follows: in Section 2, two variants of classical DE are reviewed. Section 3 introduces the PC strategy and presents the details. The experiment results on both accuracy and efficiency of PC are presented and analyzed in Section 4. Section 5 discusses the future work and concludes the paper.

#### 2. BACKGROUND: CLASSICAL DE

The classical DE algorithm can be described as a search of a real-coded vector  $\boldsymbol{x}_{n,g}$  in the search space  $\Omega$ , to optimize objective function  $f(\boldsymbol{x})$  with a dimension of D. For the first generation, in which g = 0, DE initializes candidate solutions:  $\boldsymbol{x}_{1,0}, \boldsymbol{x}_{2,0}, \cdots, \boldsymbol{x}_{N,0}$  randomly in the  $\Omega$ . Then, with constant population size of N, DE tries to improve these solutions through iterative mutation, crossover, and selection, until the termination condition is satisfied.

With an eye to the experiment in Section 4, this section mainly reviews two variants of classical DE, DE/rand/1/bin and DE/rand/1/exp.

## 2.1 DE/rand/1/bin

As its name suggests, the notation means in mutation process, each donor vector  $v_{n,g}$  is equal to a randomly chosen base individual  $x_{r_1,g}$  ("/rand/") added to one scaled difference vector ("/1/"), which is also generated from two randomly chosen individuals  $x_{r_2,g}$  and  $x_{r_3,g}$ :

$$\boldsymbol{v}_{n,g} = \boldsymbol{x}_{r_1,g} + F \cdot (\boldsymbol{x}_{r_2,g} - \boldsymbol{x}_{r_3,g})$$
 (1)

where mutation parameter  $F \in [0, 1]$  controls the significance of the difference vector built by  $\mathbf{x}_{r_2,g}$  and  $\mathbf{x}_{r_3,g}$ . The index  $r_1, r_2, r_3$  are randomly selected integers from the range [1, N], whose value differ from each other, and also differ from the target vector  $\mathbf{x}_n$ 's index n.

Moreover, the notation "/bin" refers to that in the crossover process, the number of the same elements shared by trail vector  $\boldsymbol{u}_{n,g}$  and donor vector  $\boldsymbol{v}_{n,g}$ , approximates a binomial distribution. The generation of trail vector  $\boldsymbol{u}_{n,g}$  is given by:

$$u_{d,n,g} = \begin{cases} v_{d,n,g} & \text{if } \varphi_{d,n} < Cr \\ x_{d,n,g} & \text{otherwise} \end{cases}, d = 1, 2, \cdots, D.$$
(2)

where D is the dimension of individual;  $\varphi_{d,n}$  is a uniformly distributed random number within interval [0, 1], which is independent with different d or n; crossover rate Cr is a parameter to control the probability of the components of  $\boldsymbol{x}_{n,g}$  being replaced by  $\boldsymbol{v}_{n,g}$ 's components.

Selection operator chooses the next generation's individual  $\boldsymbol{x}_{n,g+1}$  from target vector  $\boldsymbol{x}_{n,g}$  and donor vector  $\boldsymbol{v}_{n,g}$ . The criterion to weigh vectors is represented as fitness function, which is mostly equal to objective function f:

$$\boldsymbol{x}_{n,g+1} = \begin{cases} \boldsymbol{v}_{n,g} & \text{if } f(\boldsymbol{v}_{n,g}) \leq f(\boldsymbol{x}_{n,g}) \\ \boldsymbol{x}_{n,g} & \text{otherwise} \end{cases}$$
(3)

In this method, the candidate solutions can keep a constant population size N and each newly created individual  $\boldsymbol{x}_{n,g+1}$ can yield equal or better performance than former one  $\boldsymbol{x}_{n,g}$ .

# 2.2 DE/rand/1/exp

In DE/rand/1/exp, the mutation and selection operators are exactly the same Eq. (1) (3) in the DE/rand/1/bin. Only crossover method is different, in which a starting position of crossover is chosen randomly from [1, D], and Lconsecutive elements are counted in circular manner. The probability of replacing the  $d_{th}$  element from [1, L] decreases exponentially with increasing d [21].

## 3. PRINCIPAL COORDINATE STRATEGY

In this section we introduce Principal Coordinate (PC) strategy, which attempts to maximize the separability of the fitness function and get an improved DE's performance with non-separable fitness function through adaptively adjusting the coordinate system during the whole evolution process. The framework of the PC strategy is presented first, and later the method to obtain the most suitable coordinate system is discussed.

## 3.1 Framework of the PC Strategy

With the PC strategy, we suppose the best coordinate system for the fitness function is attainable. (all the coordinate systems discussed in this paper refer to orthogonal) In each generation, we draw the best coordinate system from the current population, and let crossover operator be operated under the newly-calculated best coordinate system, while without changing the coordinate system for mutation and selection operators. This is because crossover operator can generate different trail vectors under different coordinate system and has potential to improve the DE's performance [17]. One highlight of the PC strategy is the wide adaptability so it can be combined with various DE algorithms. The structure of DE with PC is briefly shown in Figure 1.



Figure 1: the PC Strategy's Framework

The following notations are important in this paper. In the D-dimension searching space  $\Omega$ , every coordinate system  $X_B$  corresponds to a basis matrix **B**. Specifically, **B**<sub>0</sub> denotes the basis matrix of the original coordinate system  $X_0$ . Obviously, **B**<sub>0</sub> = **I**. The notation of an individual under  $X_0$  can remain as  $\boldsymbol{x}_{n,g}$ , while the same individual's value under a rotated coordinate system  $X_B$  can be represented as  $\boldsymbol{x}_{n,g}(B)$ . The transformation is illustrated as

$$\begin{cases} \boldsymbol{x}_{n,g}(\boldsymbol{B}) = \boldsymbol{x}_{n,g} \cdot \boldsymbol{B}^T \\ \boldsymbol{x}_{n,g} \cdot \boldsymbol{I} = \boldsymbol{x}_{n,g}(\boldsymbol{B}) \cdot \boldsymbol{B} \end{cases}$$
(4)

where  $B^T = B^{-1}$ , which is because B is orthogonal matrix.

## **3.2** Determine the Best Coordinate System

The previous researches [12, 20] reveal that mutant vectors which generated by difference vectors are most effective mutant vectors because population distribution can exactly describe the fitness function's landscape, which is due to selection's carving effect. Based on this conclusion, we assume that the ideal offspring is highly likely to appear on the direction of difference vector. As [20] indicates, individual's stepwise movement tends to be orthogonal to the coordinate axes. Hence, we infer that the coordinate system, whose axes reflect the principal directions of difference vectors, is able to maximize the efficiency of crossover operator and improve separability of the fitness function. This assumption is preliminarily validated by our observation. Figure 2 shows empirical distributions of difference vectors under two 2-D functions. It can be seen that the best coordinate systems (arrows), which enable the fitness functions to be separable, are related to the difference vectors distributions (dots).



Figure 2: Difference Vectors Distribution and Fitness Function' Ideal Coordinate System

The algorithm to compute the best coordinate is described as below:

**Step 1.** For each individual  $x_{n,g}$ , compute three difference vectors  $\Delta$  between  $x_{n,g}$  and k nearest individuals marked as  $x_{n_1,g}, \dots, x_{n_k,g}$ .

$$\begin{cases} \boldsymbol{\Delta}_{\boldsymbol{n}_1} = \boldsymbol{x}_{n_1,g} - \boldsymbol{x}_{n,g} \\ \cdots \\ \boldsymbol{\Delta}_{\boldsymbol{n}_k} = \boldsymbol{x}_{n_k,g} - \boldsymbol{x}_{n,g} \end{cases}, n = 1, 2, \cdots, N.$$
(5)

In this paper, we set k = 3, because the significance of difference vector declines when the distance increases. So 3Ndifference vectors are obtained.

Step 2. Normalize each differential vector:

$$\boldsymbol{\Delta}_{n}^{*} = \boldsymbol{\Delta}_{n} / ||\boldsymbol{\Delta}_{n}||, \quad n = 1, 2, 3, \cdots, 3N.$$
 (6)

All the normalized differential vectors form a matrix  $\mathbf{\Delta}^* = (\mathbf{\Delta}_1^*; \mathbf{\Delta}_2^*, \cdots, \mathbf{\Delta}_{3N}^*)^T$ .

**Step 3.** Apply Principal Component Analysis (PCA) to compute the basis matrix  $B_g$  of the best coordinate system  $X_{B_g}$  for the current generation g.

$$\boldsymbol{C} = \frac{1}{3N} \boldsymbol{\Delta}^* \cdot \boldsymbol{\Delta}^{*T} \tag{7}$$

and applying SVD decomposition:

$$\boldsymbol{C} = \boldsymbol{B}_{g} \boldsymbol{D} \boldsymbol{V}^{T} \tag{8}$$

Where C is the covariance matrix of  $\Delta^*$ . The coordinate system can be transformed by  $B_g$  via Eq. (4).

The method used in Step 3 is different from classical PCA, because classical PCA reduces dimension of parameter while our method doesn't change the dimension of coordinate system, just minimizes the correlation coefficient of each coordinate axis.

The PC strategy can be combined with various DEs. Here we present the pseudo code of DE/rand/1/bin with PC as an example:

```
Initialize population: \boldsymbol{x}_{n,0}, 1 \leq n \leq N
For g \leftarrow 0 to g_{max}
t \leftarrow 1; //Counts the number of difference vectors in \Delta
For n \leftarrow 1 to N
Select 3 distinct indexes r_1, r_2, r_3;
\boldsymbol{v}_{n,g} \leftarrow \boldsymbol{x}_{r_1,g} + F(\boldsymbol{x}_{r_2,g} - \boldsymbol{x}_{r_3,g});
End For;
For i \leftarrow 1 to N
For j \leftarrow 1 to N
Computing \delta_j \leftarrow || \boldsymbol{x}_{i,g} - \boldsymbol{x}_{j,g} ||;
End For;
\delta_i \leftarrow \infty; //Avoid being selected in the next step
[w_1, w_2, w_3] \leftarrow 3 nearest vectors' indexs in \boldsymbol{\delta};
For k \leftarrow 1 to 3
egin{aligned} & \mathbf{\Delta}_t \leftarrow oldsymbol{x}_i - oldsymbol{x}_{w_k}; \ & \mathbf{\Delta}_t^* = rac{\mathbf{\Delta}_t}{||\mathbf{\Delta}_t||}; \ t++; \end{aligned}
End For; End If; End For;
//Selected the Principe Coordinate System
Getting C from \Delta^* by eq.(7);
[\boldsymbol{B}_{\boldsymbol{a}} \ \boldsymbol{D} \ \boldsymbol{H}] \leftarrow SVD(\boldsymbol{C});
Computing \boldsymbol{v}(\boldsymbol{B}_g), \boldsymbol{x}(\boldsymbol{B}_g) as eq.(4);
For d \leftarrow 1 to D
For n \leftarrow 1 to N
\mathbf{if}(rand([0,1]) < Cr) \ u(\boldsymbol{B})_{d,n,g} \leftarrow v(\boldsymbol{B})_{d,n,g};
Else u(\mathbf{B})_{d,n,g} \leftarrow x(\mathbf{B})_{d,n,g};
End If; End For; End For;
Transform \boldsymbol{u}(\boldsymbol{B}) to \boldsymbol{u} by eq.(4);
For n \leftarrow 1 to N
\mathbf{if}(f(\boldsymbol{u}_{n,g}) \leq f(\boldsymbol{x}_{n,g})) \ \boldsymbol{x}_{n,g+1} \leftarrow \boldsymbol{u}_{n,g};
Else x_{n,g+1} \leftarrow x_{n,g};
End If; End For; End For; //End
```

## 4. EXPERIMENTS

This section aims at evaluating the PC strategy in two aspects: accuracy and efficiency. Accuracy is the level that the coordinate system obtained by PC is close to the ideal coordinate system which maximizes the separability of the fitness function. Efficiency means the algorithm's convergence speed to the optimal solution.

The PC strategy is independent from mutation, crossover and selection operators, so it can be combined with different DE's variants. Without loss of generality, in the experiments two variants of classical DE, DE/rand/1/bin and DE/rand/1/exp are chosen to combine with the PC strategy, which respectively correspond to the notation DE/rand/1/bin /PC and DE/rand/1/exp/PC. These algorithms are compared with the original DEs to test PC's ability.

## 4.1 Accuracy of the PC strategy

In this experiment, DE/rand/1/bin/PC and DE/rand/1/ exp/PC are respectively applied to two benchmark functions, Rosenbrock Function and rotated Michalewicz Function under 2-D searching space. The angle, by which the original coordinate system clockwise rotate to a specific coordinate system, can be used to uniquely define the coordinate system. Rosenbrock Function and Michalewicz Function are chosen because their ideal coordinate systems are already known. Rosenbrock Function is a non-separable function, but it becomes approximate separable when the coordinate system is rotated by 45°. Michalewicz Function's ideal coordinate is  $X_0$  since it is separable. In order to make this experiment more convincing, the coordinate system of Michalewicz Function is rotated by  $\theta = 35^{\circ}$ . Therefore the best rotation angle for this function also changes to 35°.

The experiment results are shown in Figure 3, which records



Figure 3: The Rotation Angle Obtained by The PC Strategy

the changes of rotation angles obtained by the PC strategy over generations.

From Figure 3, we can see that before the  $6_{th}$  generation in Rosenbrock Function and  $20_{th}$  generation in Michalewicz Function, the rotation angles obtained by PC have big dithering, which is because the difference vectors are gotten from relative random population distribution. As the generations' number increases, the difference vectors have collected enough information about the fitness function, so the angle gets steadier and approximates to the ideal angle. In the last several generations, the angle might become unsteady again, which is caused by that the population is gathered in the optimal solution and the difference vectors get less relevant. This experiment's results indicate that the PC strategy has good ability to draw the best coordinate system from the current population and has robustness to overcome many local optima in Michalewicz Function.

#### 4.2 Efficiency of the PC Strategy

In this experiment, we present a comparison between classical DEs and the combination of DEs and the PC strategy in algorithms' performance. DE/rand/1/bin, DE/rand/1/exp, DE/rand/1/bin/PC and DE/rand/1/exp/PC are tested by a set of benchmark functions as well as an engineering project.

#### 4.2.1 Benchmark Functions

Four benchmark functions  $f_1(\mathbf{x}) - f_4(\mathbf{x})$  are given in Table 1, among which  $f_1(\mathbf{x})$  contains gaussian noise;  $f_2(\mathbf{x})$ ,  $f_3(\mathbf{x})$ ,  $f_4(\mathbf{x})$  are multimodal;  $f_1(\mathbf{x})$ ,  $f_2(\mathbf{x})$ ,  $f_4(\mathbf{x})$  are non-separable; and all functions here are rotated by a random rotation matrix.

Set parameters as follow: N = 50, Cr = 0.6, F = 0.85. In this experiment, we apply each algorithm in every function

Table 1: The Benchmark Functions

Function	Domain
$f_1(\boldsymbol{x}) = \left(\sum_{i=1}^{10} \left(\sum_{j=1}^i z_j\right)^2\right) \cdot (1 + 1)$	[-100, 100]
0.4 N(0,1) )	
$f_2(\boldsymbol{x}) = -20 \exp(-0.2 \sqrt{\frac{1}{10} \sum_{i=1}^2 z_i^2}) -$	[-100, 100]
$\exp(\frac{1}{10}\sum_{i=1}^{10}\cos(2\pi z_i)) + 20 + e$	
$f_3(\boldsymbol{x}) = -\sum_{i=1}^5 \sin(z_i) \cdot \left(\sin\left(\frac{i \cdot z_i^2}{\pi}\right)\right)^{20}$	$[-\pi,\pi]$
$f_4(\boldsymbol{x}) = \sum_{i=1}^{10-1} \left( 100(z_i^2 - z_{i+1})^2 + (z_i - 1)^2 \right)$	[-100, 100]
$z_i = (x_i - o) \cdot \boldsymbol{M}$	

for 10 times, from which we compute the mean number of generations when a set of value-to-reach (VTR) terminating conditions are satisfied, shown as Figure 4. The successful rate of algorithm, which means the rate to find the optimal solution in the experiment, is also analyzed.

From the results obtained in four benchmark functions, we can see that the PC strategy generally improves the efficiency of the original classical DEs. Given VTR=1e - 5, in  $f_1(\boldsymbol{x})$  (Schwefel Function with Noise), PC reduces DE/rand /1/exp's generations by 7.2%. Nevertheless, PC does not bring much advantage to DE/rand/1/bin, which is because the instability of coordinate system obtained by PC in the beginning of experiment keeps the algorithm performing better.

In  $f_2(\boldsymbol{x})$  (Ackley's Function), which is a non-separable function with strong multimodal behavior, PC improves the efficiency of DE/rand/1/bin by 1.07% and DE/rand/1/exp



Figure 4: The Comparison Between the Hybridized Algorithms and Classical DEs under Benchmark Functions

10.02%, which indicates that PC has enough robustness to overcome the deception by local optima.

In the experiment of  $f_3(\mathbf{x})$  (Michalewicz Function), PC brings significant improvements to both DE/rand/1/bin and DE/rand/1/exp: a reduction of 7.03% in DE/ rand/1/bin's generation number and 10.74% in DE/rand/1/exp's.

The experiment result of  $f_4(\mathbf{x})$  (Rosenbrock Function) shows that classical DEs gain more stability when combined with PC. The figure reads DE/rand/1/bin's performance is much worse than DE/rand/1/exp. However, hybridized with PC, DE/rand/1/bin's efficiency is improved by 59.53%, getting a performance almost as good as DE/rand/1/exp, and DE/rand/1/exp's generation number is also reduced by 3.17% with PC.

Besides, PC brings DEs higher successful rates. In  $f_1(\mathbf{x})$ ,  $f_2(\mathbf{x})$ ,  $f_4(\mathbf{x})$ , successful rates in our experiment are 1, while in  $f_3(\mathbf{x})$ , the original classical DEs get a low successful rate: DE/rand/1/bin 32% and DE/rand/1/exp 34%, which is due to the strong multimodal behavior. In  $f_3(\mathbf{x})$ , PC improves the successful rate to: DE/rand/1/bin/PC 52% and DE/rand/1/exp/PC 59%, (from 400 times experiments)which proves that PC increases classical DEs robustness for multimodal fitness function.

In sum, PC is able to improve DE's efficiency, stability and robustness.

#### 4.2.2 Engineering Project: Sidelobes Suppression

Sidelobes suppression is a practical problem in telecommunication [8]. In optical communication, the optimization goal is to maximize the ratio of mainlobe's radiation pattern and the maximum sidelobe's radiation pattern, which is illustrated by Figure 5. The objective G(w) function is defined as

$$G(\boldsymbol{w}) = \min\left\{-10 \lg \frac{|E(\theta = 0, \boldsymbol{w})|}{\max_{\varepsilon < |\theta| < \frac{\pi}{2}} |E(\theta, \boldsymbol{w})|}\right\}$$
(9)

$$E(\theta, \boldsymbol{w}) = \sum_{d=1}^{D} \frac{\sin(\frac{\pi}{2\lambda}\sin\theta)}{\frac{\pi}{2\lambda}\sin\theta} \exp(jk\theta \sum_{j=1}^{d} w_j) \qquad (10)$$

Where  $w_d$  is the width of the  $d_{th}$  waveguide;  $\lambda$  denotes wavelength and k wave number;  $\varepsilon$  is width of mainlobe.

DE has been one of the classic methods to this optimization problem. In the experiment, the parameters are set as: N = 50, f = 0.85, Cr = 0.9, D = 30. The fitness values for the first 300 generations are recorded, shown in Figure 6. It can be seen that PC significantly accelerates the convergence of classical DEs. Taken fitness value=-9.5 as VTR, PC reduces 61.5% generations of DE/rand/1/bin and 77.2% of DE/rand/1/exp, while PC only adds 7.8% cost to classical DEs per generation, which is 0.005s of extra time. As a result, PC reduces the cost by 61.6% in DE/rand/1/bin and 80.1% in DE/rand/1/exp. This result indicates that PC has good potential in engineering application, and its extra cost per generation is minor when compared with the efficient improvement it brings.



Figure 5: The Illustration of Sidelobes Suppression in Optical Communication



Figure 6: The Fitness Value Over Generations

# 5. CONCLUSION & FUTURE WORK

In this paper, the PC strategy, a novel adaptive control strategy for EA is proposed. This approach constructs an adaptive method to adjust the coordinate system during the whole evolution process. The Accuracy Experiment shows that PC can efficiently draw approximate ideal coordinate system from the current population distribution. Efficiency Experiments suggests that PC improves efficiency, stability and robustness of classical DEs. Lastly, the real scenario application proves PC's advantage for complicated engineering application.

As for future research, the authors intend to broaden the matrix conversion from rotation transformation to linear transformation, to further maximize the separability of function. The authors also plan to explore a steadier and more effective way to draw the ideal coordinate system in high dimension's cases.

## 6. **REFERENCES**

- H. Abdi and L. J. Williams. Principal component analysis. Wiley Interdisciplinary Reviews: Computational Statistics, 2(4):433–459, 2010.
- [2] D. L. Battle and M. D. Vose. Isomorphisms of genetic algorithms. Artif. Intell., 60:155–165, March 1993.
- [3] D. Beasley, D. R. Bull, and R. R. Martin. A sequential niche technique for multimodal function optimization. *EVOLUTIONARY COMPUTATION*, 1:101–125, 1993.
- [4] J. Brest, S. Greiner, B. Boskovic, M. Mernik, and V. Zumer. Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems. *IEEE Trans. Evolutionary Computation*, 10(6):646–657, 2006.
- [5] S. Das and P. N. Suganthan. Differential evolution: A survey of the state-of-the-art. *IEEE Transactions on Aerospace and Electronic Systems*, 15(1):27–54, February 2011.
- [6] M. Dorigo, V. Maniezzo, and A. Colorni. Ant system: optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, 26(1):29–41, Feb. 1996.
- [7] H. Y. Fan and J. Lampinen. A trigonometric mutation operation to differential evolution. J. of Global Optimization, 27:105–129, Sept 2003.

- [8] M. Gregory, J. Petko, T. Spence, and D. Werner. Nature-inspired design techniques for ultra-wideband aperiodic antenna arrays. *Antennas and Propagation Magazine, IEEE*, 52(3):28–45, june 2010.
- [9] J. Hertz, R. G. Palmer, and A. S. Krogh. Introduction to the Theory of Neural Computation. Perseus Publishing, 1991.
- [10] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by Simulated Annealing. *Science*, 220(4598):671–680, May 1983.
- [11] W. Langdon and R. Poli. Evolving problems to learn about particle swarm and other optimisers. In Evolutionary Computation, 2005. The 2005 IEEE Congress on, volume 1, pages 81–88 Vol.1, sept. 2005.
- [12] K. V. Price. An introduction to differential evolution, pages 79–108. McGraw-Hill Ltd., UK, Maidenhead, UK, England, 1999.
- [13] A. K. Qin, V. L. Huang, and P. N. Suganthan. Differential evolution algorithm with strategy adaptation for global numerical optimization. *IEEE Trans. Evolutionary Computation*, 13(2):398–417, 2009.
- [14] J. Rönkkönen, S. Kukkonen, and K. Price. Real-parameter optimization with differential evolution. In *Congress on Evolutionary Computation* [14], pages 506–513.
- [15] J. E. Rowe, M. D. Vose, and A. H. Wright. Group properties of crossover and mutation. *Evol. Comput.*, 10:151–184, June 2002.
- [16] J. E. Rowe, M. D. Vose, and A. H. Wright. Structural search spaces and genetic operators. *Evol. Comput.*, 12:461–493, Dec 2004.
- [17] R. Salomon. Reevaluating genetic algorithm performance under coordinate rotation of benchmark functions - a survey of some theoretical and practical aspects of genetic algorithms. *BioSystems*, 39:263–278, 1995.
- [18] R. Storn and K. Price. Differential evolution a simple efficient adaptive scheme for global optimization over continuous spaces. Technical Report 95-012, Int. Compt. Sci. Inst., Berkeley, CA, 1995.
- [19] R. Storn and K. Price. Differential evolution: A simple and efficient heuristic for global optimization over continuous spaces. *IEEE Transactions on Aerospace* and Electronic Systems, 11:341–359, December 1997.
- [20] A. M. Sutton, M. Lunacek, and L. D. Whitley. Differential evolution and non-separability: using selective pressure to focus search. In *Proceedings of the* 9th annual conference on Genetic and evolutionary computation, GECCO '07, pages 1428–1435, New York, NY, USA, 2007. ACM.
- [21] J. Tvrdik. Adaptive differential evolution and exponential crossover. In Computer Science and Information Technology, 2008. IMCSIT 2008. International Multiconference on, pages 927 –931, oct. 2008.
- [22] D. Zaharie. Extensions of differential evolution algorithms for multimodal optimization. In In Proceedings of SYNASCaŕ04, 6th International Symposium of Symbolic and Numeric Algorithms for Scientific Computing, pages 523–534, 2004.