*2009 International Conference on Machine Learning and Computing*
*IPCSIT vol.3 (2011) © (2011) IACSIT Press, Singapore*

# Improving Differential Evolution Algorithm with Activation Strategy

Zhan-Rong Hsu [1 +], Wei-Ping Lee [1], Ching-Wei Chien [1]

[1] Chung Yuan Christian University, Information Management Department, Chun li, Taiwan, 32023

**Abstract.** Evolutionary Computation (EC) provides high performance on real world optimization problems such as scheduling, resource distribution, portfolio optimization etc. Differential Evolution (DE) algorithm was first reported in 1995. Based on the characteristics of simple structure, high accuracy and efficiency, and the requirement of fewer parameters, DE has received significant attention from researchers. It has been applied to numerous fields and performs much better than other evolutionary computation.Although DE shows powerful performance, but it has attach importance to the drawbacks of unstable convergence, breakaway the solution space, and the common defect of evolution computation "dropping into regional optimum". In this study, inspired from Genetic Algorithm (GA), we attempt to improve the traditional differential evolution algorithm and propose a novel algorithm "Activated Strategy Differential Evolution" (ASDE). Based on import the Activated Strategy (AS) intensified the structure of traditional DE for enhancing the accuracy and efficiency again

**Keywords:** Evolution Computation (EC), Differential Evolution (DE), Genetic Algorithm (GA), Activated Strategy Differential Evolution (ASDE), Activation Strategy (AS).

## 1. Introduction

In real world, every problem has different "Feasible solutions", and these feasible solutions will be in accordance with the needs of different individuals have different restrictions, such multi-solution problems called NP-Hard problems namely not yet to found the appropriate solution's problems.

Recently, the focus of evolution computation was on different types of optimization problems such as multi-variable, highly non-linear and non-differential problems. Since 1995, a novel algorithm "Differential evolution" was announced, with high accuracy and efficiency.

In comparison with Particle Swarm Optimization (PSO), Genetic Algorithm (GA) and else Evolutionary Computation (EC), DE provides more powerful performance. Nevertheless, there exist some problems like dropping in regional optimum, break away solution space and unstable convergence in the case of DE. In this study, to enhance the capacity of DE, we proposed Activated Strategy Differential Evolution (ASDE), import activation strategy to DE effect upon disturb the algorithm to give activation efficacy for enhancing the accuracy of solution. This study will examine several benchmark functions to evaluate the performance of ASDE to confirm that import activated strategy can help increase the performance of basic DE.

## 2. Literature Review

### 2.1. Evolutionary Computation (EC)

Evolutionary computation is one of the important aspects in artificial intelligence, the main idea inspired by Darwin 『Theory of evolution』. Nature is a complex system constructed by individuals accompanied by the environment. The complex system will transform and adapt gradually to gain more opportunity survival [1]. Therefore, organisms in the nature use co-evolution to generate high adaptability organisms for adapting the environment under continuous change [9].

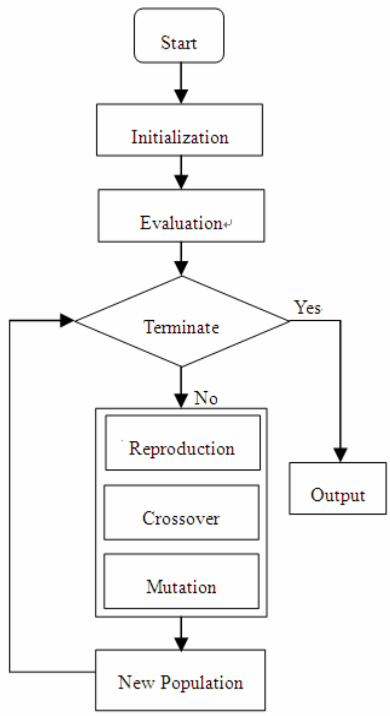
+ Corresponding author. Tel.: +886955753026 ; fax: +88632655499.
*E-mail address*: gackt@gmail.com.

Two different perspectives to explain the evolution computation, a heuristic computation inspired by community-based animals and then use it to find the optimum solution for different portfolio optimization problems.

From the perspective of biology, the individual of population has to compete for survival. During the competition, the loser will be eliminated and the winner will get a chance to exist and reproduce [2] [3]. From the perspective of numerical analysis, evolution computation is an optimization algorithm similar to random search. The essence of evolution computation is non-deterministic and, therefore, can explore the feasible solution in the solution space. In the evolution computation, numerous individuals will continuously follow the "niche” and each individual can be regarded as a solution for the problem. This is the reason why evolution computation has the diverse capacity to solve problems. Most complex non-linear functions lack the efficiency to seek appropriate solution. Hence, evolution computation has become the most effective tool for numerical analysis [2].

## 2.2. Genetic Algorithm (GA)

Genetic algorithm inspired from Darwin “Nature Selection” by Holland [3], furthermore simulate Biological adaption and evolution for complex environment moreover applied to artificial intelligence system. In 1989, Goldberg investigated the mechanism of genetic algorithm systematically using the basic operators “reproduction”, “Crossover”, “Mutation”. GA is one of the search rules based on the theory of natural selection and exercise simulation by a computer. First, decide initial population, simulate the evolution and adaption mechanisms and use the GA operator to generate individuals. During the generation, the most suitable individual (appropriate solution) will be retained at the end [3, 9, 12]. The procedure of GA is described in the following.




The Genetic Algorithm procedure is following:

(1) Reproduction：

In order to identify and replace unsuitable parents, this mechanism will identify better chromosomes with high probability. The fitness function will evaluate whether the chromosomes should be reserved or not. If the initial population includes N chromosomes, the reproduction mechanism will reproduce the same number of chromosomes [11].

(2)Crossover：

Different chromosomes are enabled through the crossover mechanism and the information is exchanged randomly. That is two chromosomes are combined according to the crossover function to compose a new chromosome with the characteristics of parents. The purpose is to combine a new chromosome provider with better fitness, but there is a risk of inheriting bad characteristics from parents; therefore, crossover mechanism cannot confirm whether new chromosomes are always better than their parents [11].

(3)Mutation：

During the evolution process, individuals will mutate the segment of genetics to enhance the fitness to adapt variation. Mutation mechanism is to prevent premature convergence, exploration of new solution space, and to avoid dropping into regional optimum [11].

## 2.3. Differential Evolution (DE)

Differential evolution was reported in 1995, which is derived from the Price for solving Chebyshev polynomial fitting problem [4]. This was formally proposed in 1996 during the first session of IEEE evolution optimization conference, and become one of the well known algorithm [5,6,7]. Differential evolution constructed on individual difference, added the difference to another and observed whether bring positive result or not. According to the concept, obtain the purpose of evolution.

The Differential evolution procedures and parameters setting are described in the following:

(1)Initialization: Setup the related parameters.

(2)Mutation: Three vectors $X_{r1,G}$、$X_{r2,G}$、$X_{r3,G}$ are random selected, and the Donor Vector ($V_{i,G+1}$) is generated according to the following formula and mutation factor ($F$) as $V_{i,G+1} = X_{r1,G} + F\,(X_{r2,G} - X_{r3,G})$

(3)Crossover: Crossover the Donor Vector and the Target Vector ($X_{i,G}$) selected from the population to generate the Trial Vector ($U_{i,G}$). The following formula decides the Crossover rate. In the generation of $j$, the ith segment of $U_{i,G}$ is composed by $X_{i,G}$ or $V_{i,G}$ .Illustrate with Fig 2.

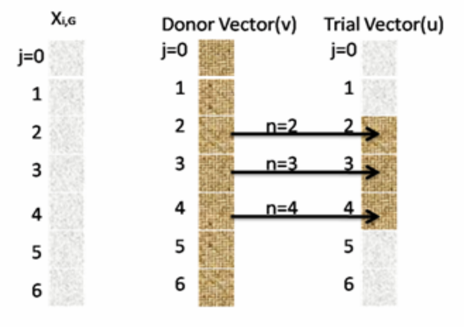



Fig. 2 Crossover mechanism [6]

(4) Selection: Compare with the Trial Vector ($U_{i,G}$) and Target Vector ($X_{i,G}$) to decide the next generation new Target vector.

The parameter setting is described in the following:

(1)Number of Vector (NP):

The total number of the solution vector in the generation is determined based on the Number of Population. Large population required longer computing time; on the contrary, small population in search of a large solution space may not be very efficient.

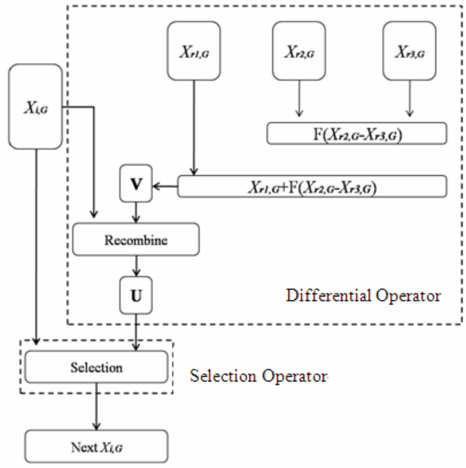



Fig. 3 The procedures of Differential Evolution

(2)Mutation factor ($F$):

This factor decides how many perturbation ratios the solution can acquire. If the value is larger, the magnitude of jump out will increase, namely obtain better ability to breakaway regional optimum, but the efficiency of convergence will be decelerate. On the other hand, smaller mutation factor can converge rapidly, but has a relatively high probability of dropping into the regional optimum. Generally, F suggests setting between 0 and 2 (But not 0). Furthermore, according to the previous studies, F is generally set between 0.4 and 1. But is not yet conclusive, the F value should be investigate in depth [9, 12].

(3) Crossover Rate (CR):

Determine the swapping probability between the trial vector and target vector. Larger CR means more segment should be swapped between the two vectors, in other words, both the trial vector and target vector are highly similar.

Differential evolution can be separated to "differential operator" and "selection operator". Illustrate with Fig. 3, first random select 4 solution vectors, one target vector else vectors Xr1,G , Xr2,G , Xr3,G get into mutation mechanism. After mutation mechanism, obtain a new vector called Donor Vector (Vi,G). Furthermore according to combination to generate Trial Vector (Ui,G). Eventually according to Selection operator to select which can be reserved into next generation? The computation formula is following: Vi,G+1 =Xr1,G + F (Xr2,G − Xr3,G)

## 2.4. Activation Strategy (AS)

For activation, look for the first non-zero value in the solution vector and randomly alternates by 0-9. If select the segment located at bottom after decimal point in solution vector will cause insufficient for perturbation. The purpose to look for the first non-zero value is avoided in the above situation to enhance the evolution performance as illustrated in Fig 4. Before perturbation, the solution vector was 0.0004512673193, after perturbation it became 0.0007512673191.Taking the random value for replacing the original value to perturb solution vector, the activation strategy can help the solution vector break away when the solution vector drops into the regional optimum. But to a certain extent may not breakaway region optimum effectively, but through

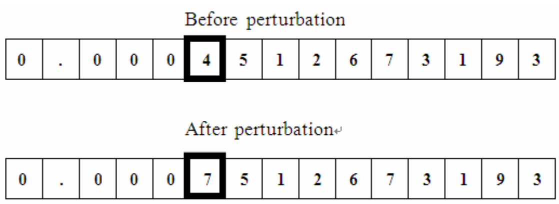



Fig. 4 Activation Strategy

selection mechanism eliminate unsuitable solution to avoid unsuitable solution reserved into next generation caused performance to decrease. In this study, importing activation strategy to selection operator for giving another mutation cause effectively exploration solution space, furthermore giving the ability of perturbation prevent to drop into region optimum cannot break away. Simultaneously improve the common problems such as unstable convergence and dropping into region optimum.

# 3. Research Methods and Design

## 3.1. Experimental Design

In this study, we proposed a novel algorithm by importing Activation Strategy (AS) to improve DE/best. Approach to implement activation strategy to give second chance mutate, namely offer more alternative solution for the selection mechanism to select best one into next generation. During the generation expansion, approach to implement above structure will optimize the solution in every generation for increasing convergence rate. Therefore in early period, ASDE will represents powerful convergence rate, according to provide with more alternative solution for selecting can make convergence more stable. The improvement procedures and flow chart is following:

Step 1: Initialization; Initialize all the parameters.
Step 2: Random select initial vectors.
Step 3: Evaluate fitness.
Step 4: Mutation; According to the computation formula, generate the donor vector.
Step 5: Recombination; Crossover the Donor vector and Target Vector to generate the trial Vector.
Step 6: Selection; compare it with the Activation Trial Vector and the original Target Vector and Trial Vector. Furthermore determine the one that can be reserved for the next generation.
Step 7: Upon achieving the goal, terminate the condition or back to step 2.

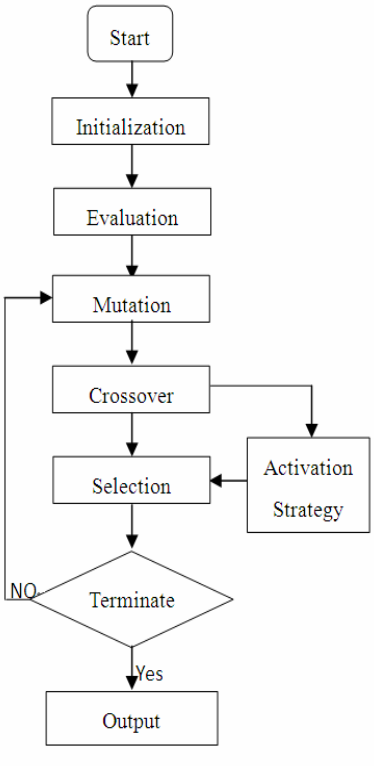


Fig. 5 The flow chart of ASDE

## 3.2. Parameter setup and Experimental Benchmark Function

The experimental parameter setting is discussed in the Table I. In this study, utilize four benchmark functions for evaluating the performance of Activation Strategy Differential Evolution (ASDE). All of the benchmarks will be separate repeated 1000 times (30-Dimension, 1000 generation) and 100 (60-Dimension, 2000 generation) times to obtain the average. The DE/rand and DE/best are compared to understand ASDE's effectiveness. The functions characteristics show in Table II.

TABLE 1.
THE TABLE OF PARAMETER SETUP

| | |
|---|---|
| Mutation Factor(*F*) | 0.5 |
| Crossover rate (*CR*) | 0.9 |
| Maximum of iteration(*iter*) | 1000,2000 |
| Upper bound | 100 |
| Lower bound | -100 |
| Number of vector(*NP*) | 50 |
| Dimension (*Dim*) | 30, 60 |

TABLE 2.
Function characteristics

| Number | Name | Category | Function |
|---|---|---|---|
| *f1* | *Sphere* | Single-modality | $f(x) = \sum (x_i)^2$ |
| *f2* | *Rosenbrock* | Single-modality | $f(x) = \sum [100*(x_i^2 - x_{i+1})^2 + (1-x_i)^2]$ |
| *f3* | *Griewank* | Multi-modality | $f(x) = \sum \frac{x_i^2}{4000} - \Pi \cos(\frac{x_i}{\sqrt{i}}) + 1$ |
| *f4* | *Rastrigrin* | Multi-modality | $f(x) = \sum (x_i^2 - 10*\cos(2\pi x_i) + 10)$ |

# 4. Experimental Results

In this study, DE/rand, DE/best, and ASDE on the performance of benchmark functions Sphere, Rosenbrock, Griewank, Restrigrin are compared. The four benchmark functions had a theoretical minimum value of 0. In the experimental study, we compared DE/rand, DE/best and ASDE to confirm with their own stability and the best experimental result, and ultimately through the experimental result analysis their performance in different dimensions complex function.

■ Benchmark function: Sphere (f1)

TABLE 3. Experiment Result of Sphere (*f1*)

| f1 / Dimension | | DE/rand | DE/best | ASDE |
|---|---|---|---|---|
| Avg | 30 | 3.645099e-10 | 3.037896e-16 | 2.863411e-21 |
| | 60 | 1.08362e-11 | 1.880908e-19 | 4.968614e-24 |
| Best | 30 | 1.382495e-10 | 9.706391e-17 | 6.649855e-22 |
| | 60 | 5.421986e-12 | 5.89369e-20 | 1.06073e-24 |

TABLE 4. Experiment Result of Rosenbrock (*f2*)

| f2 / Dimension | | DE/rand | DE/best | ASDE |
|---|---|---|---|---|
| Avg | 30 | 1.196372e+02 | 7.177397e+1 | 5.532458e+1 |
| | 60 | 1.195179e+02 | 1.36946 e+02 | 1.211944e+2 |
| Best | 30 | 3.99199e+01 | 6.405021e+0 | 6.860276e+0 |
| | 60 | 6.203696E+01 | 2.934326e+01 | 4.813753e+1 |

According to the experiment result and convergence progress, ASDE in the single-modality function Sphere provides significant performance than DE/best and DE/rand. Therefore we can confirm ASDE is more effectiveness. ASDE represents more powerful solving ability than original DE.

■ Benchmark function: Rosenbrock (f2)

In the benchmark function Rosenbrock (f2), ASDE has slight difference with DE/best in “best” value, but average is better than DE/best between 30 and 60 dimension. According to the result, in function Rosenbrock, ASDE shows more stability than improvement target DE/best.

■ Benchmark function : Griewank(f3):

In this function, DE/rand is much better than DE/best in average, therefore, in this function DE/best (improvement object) has poor performance. After importing activation strategy, ASDE to give consideration to stability and performance, moreover significant exceed DE/rand. In different dimension, ASDE still provides superiority performance, ASDE obtain the theoretical minimum value of 0 regardless Avg or Best in 30 and 60 dimensions.

■ Benchmark function : Rastrigrin(f4):

In Restrigrin (f4), compared with these algorithms, ASDE is significantly superior in stability and performance. In the later period of convergence, ASDE has outstanding gap with DE/rand and DE/best in 30 dimensions. According to the experimental result, importing the activation strategy, ASDE proof the performance and stability in this function. But when the dimension extend, ASDE likely to be affected by improvement object DE/best.

TABLE 5. Experiment Result of Griewank (*f3*)

| f3 / Dimension | | DE/rand | DE/best | ASDE |
|---|---|---|---|---|
| Avg | 30 | 6.180916e-10 | 7.196729e-4 | 0 |
| | 60 | 8.298739e-13 | 7.39604e-3 | 0 |
| Best | 30 | 3.212564e-11 | 0 | 0 |
| | 60 | 1.781908e-13 | 0 | 0 |

TABLE 6. Experiment Result of Rastrigrin (*f4*)

| f4 / Dimension | | DE/rand | DE/best | ASDE |
|---|---|---|---|---|
| Avg | 30 | 3.196104e+0 | 5.173807e-1 | 1.535211e-10 |
| | 60 | 1.019117e+2 | 3.430157e+1 | 2.634736e+1 |
| Best | 30 | 5.712408e-1 | 5.866465e-8 | 1.241673e-12 |
| | 60 | 7.614687e+1 | 5.470902e+0 | 1.313843e+1 |

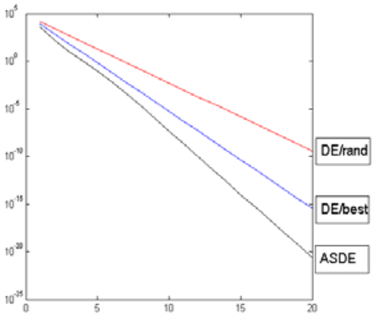



Fig. 6 *Sphere* (*f1*) 30-Dimension

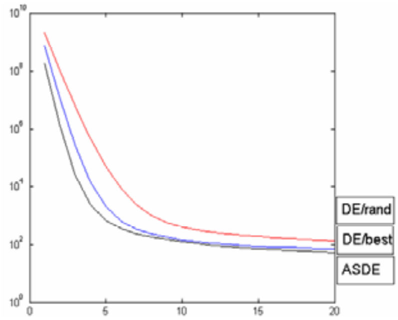



Fig. 7 Rosenbrock (*f2*) 30-Dimension

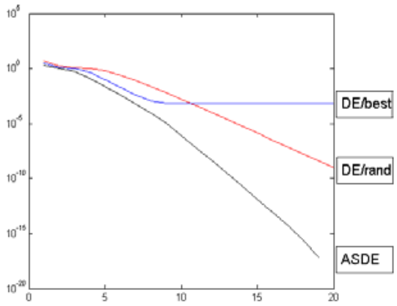



Fig. 8 *Griewank* (*f3*) 30-Dimension

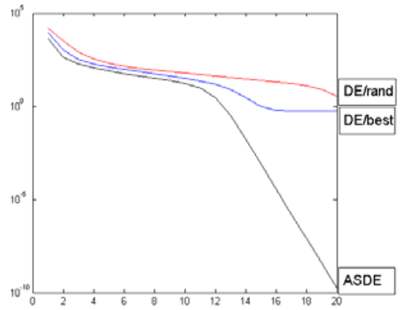



Fig. 9 *Rastrigrin (f4)* 30-Dimension

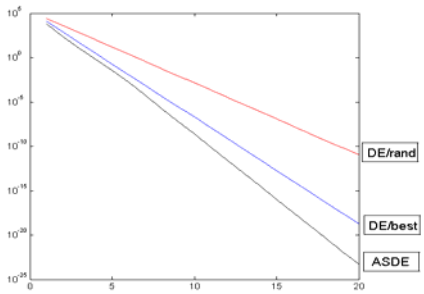



Fig. 10 *Sphere* (*f1*) 60-Dimension

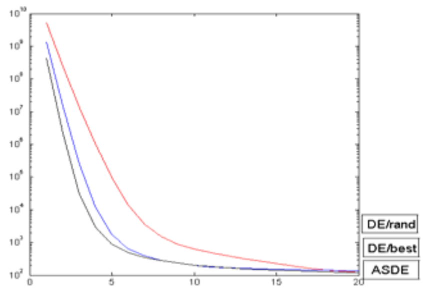



Fig. 11 *Rosenbrock* (*f2*) 60-Dimension

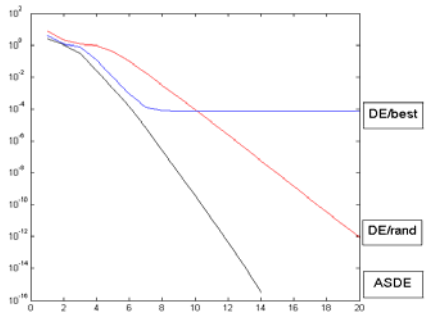



Fig. 12 *Griewank* (*f3*) 60-Dimension

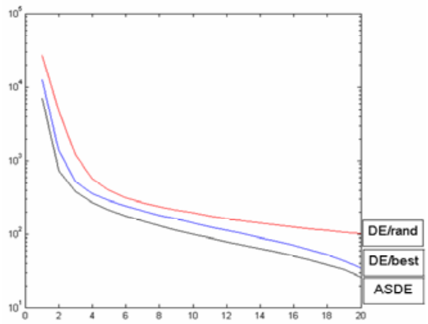



Fig. 13 *Rastrigrin (f4)* 60-Dimension

## 5. Conclusion

In this study, we depend on the import activation strategy to improve the DE/best to enhance the performance and stability. Eventually we proposed a novel algorithm "ASDE" which can increase the capacity of exploration solution space, simultaneously providing the ability of perturbation to avoid dropping into regional optimum. Approach to this new algorithm "ASDE", reform the evolutionary computation common problems and to enhance the differential evolution performance newly.

According to the experimental results, ASDE about performance and convergence has been confirming have significant effect, especially multi-modality function. ASDE not only improves the drawbacks of traditional DE such as unstable convergence and easy to drop into regional optimum, more proof this novel algorithm provide with high stable convergence and significant performance than original DE, but also the first study implement activation strategy into differential evolution algorithm. In the future, we will investigate different activation strategies constantly to enhance performance for solving more complex function and real-life application problems.

## 6. References


[1] P. J. Bentley, T. Gordon, J. Kim and S. Kumar, "New trends in Evolutionary Computation," *The 2001 Congress on Evolutionary Computation*, 2001.

[2] H.–P. Schwefel, "Numerical Optimization of Computer Models," *Wiley*, Chichester, 1981.

[3] D. E. Goldberg, "Genetic Algorithms in Search Optimization and Machine Learning," *Addison-Wesley*, 1989.

[4] K. Price, "Differential Evolution: A Fast and Simple Numerical Optimizer", *1996 Biennial Conference of the North American Fuzzy Information Processing Society,* June 1996, pp.524-527.

[5] R. Storn, "On the usage of differential evolution for function optimization", *1996 Biennial Conference of the North American Fuzzy Information Processing Society,* June 1996, pp.519–523.

[6] R. Storn and K. Price, "Minimizing the real functions of the ICEC'96 contest by Differential Evolution", *Int. Conf. on Evolutionary Computation, Nagoya, Japan,* 1996, pp. 842-844.

[7] M. M. Ali, "Differential evolution with preferential crossover," *European journal of operational research,* 2007, pp.1137-1147.

[8] H.-K. Kim, J.-K. Chong, K.-Y. Park and D. A. Lowther, "Differential Evolution Strategy for Constrained Global Optimization and Application to Practical Engineering Problems," *IEEE Transactions on Magnetics,* vol.43, 2007, pp. 1565-1568.

[9] W. M. Spears, K. A. De Jong, T. Back, D. B. Fogel and H. deGaris, "An overview of evolutionary computation," *Proceedings of the 1993 European Conference on Machine Learning*, 1993.

[10] Th. Back, F. Hoffmeister and H.-P. Schwefel, "A survey of evolution strategies," *Proceedings of the Fourth International Conference on Genetic Algorithms, Morgan Kaufmann, San Mateo*, 1991, pp. 2-9.

[11] Th. Back and H.-P. Schwefel, "An overview of evolutionary algorithms for parameter optimization," *Evolutionary Computation*, vol.1 no.1, 1993, pp. 1-24.

[12] H. Muhlenbein, G. Schleuter and O. Kramer, "Evolution algorithms in combinatorial optimization," *Parallel Computing,* vol.7, 1988, pp. 65-88.