Tuning of an Adaptive Unified Differential Evolution Algorithm for Global Optimization

Ji Qiang and Chad Mitchell
Lawrence Berkeley National Laboratory
Berkeley, USA
Email:jqiang@lbl.gov, chadmitchell@lbl.gov

Albert Qiang
Campolindo High School
Moraga, USA
Email: albertqiang1@gmail.com

***Abstract*—Recently, an adaptive unified differential evolution algorithm for single-objective global optimization was proposed in an internal report. Instead of the multiple mutation strategies in conventional differential evolution algorithms, this algorithm employs a single equation unifying multiple strategies into one expression. It has the virtue of mathematical simplicity and also provides users the flexibility for broader exploration of the space of mutation operators. However, the four control parameters used in the unified mutation expression might slow down the speed of convergence due to an over exploration of the search space. In this paper, we systematically studied the choice of those control parameters in the unified mutation strategy using fourteen unimodal and multimodal functions from the CEC2005 benchmark. Those numerical results suggest that the use of three control parameters in the unified mutation strategy improves the performance of the original algorithm and shows promising performance in comparison to several conventional differential evolution algorithms.**

I. Introduction

Global optimization has many applications in science, engineering, economics, and social life. It is also a challenging area of research since the problem to be solved can be continuous or discrete, smooth or noisy, unimodal or multimodal, constrained or unconstrained. Evolutionary algorithms (EAs) use a stochastic optimization approach and have been successfully applied to a variety of global optimization problems. Many evolutionary algorithms such as genetic algorithms (GA) [1], evolution strategies (ES) [2], evolutionary programming (EP) [3], particle swarm optimization (PSO) [4], and differential evolution (DE) [5], have been studied for global optimization. Among these methods, differential evolution is a simple yet efficient population-based, stochastic, evolutionary algorithm. It was first introduced by Storn and Price in 1995 to optimize real parameter, real-valued functions and has generated much interest since then [5], [6], [7], [8], [10], [11], [12]. In a number of studies, the differential evolution algorithm performed effectively in comparison to several stochastic optimization methods such as simulated annealing, controlled random search, evolutionary programming, the particle swarm method, and genetic algorithms [6], [13], [14], [15]. It has been successfully used in a variety of applications and demonstrated its effectiveness.

The differential evolution algorithm uses the scaled differences of parent solutions to form a mutation operator to generate next-generation candidates for global optimization. In the paper of Storn and Price, five different mutation strategies were proposed [6]. Several additional variants of these mutation strategies were later proposed to improve the properties of the mutation operation, e.g. to make it rotationally invariant [16]. The presence of multiple mutation strategies makes the differential evolution algorithm complicated to implement and use appropriately. In some recent studies [17], [18], [19], algorithms using a pool of several mutation strategies were also proposed. The choice of a specific strategy in these algorithms was mostly based on qualitative properties of the mutation strategy and empirical experience. There is no systematic method that automatically decides which mutation strategy should be used. Recently, we proposed an adaptive unified differential evolution (AuDE) algorithm for global optimization in an internal report [20]. The uniqueness of this algorithm lies in the use of a single mutation expression that encompasses almost all commonly-used mutation strategies as special cases. This is mathematically simpler than the conventional algorithm with its multiple mutation strategies, and also provides users the flexibility to explore new combinations of conventional mutation strategies during optimization. By making the control parameters in the mutation and crossover stages self-adaptive, different mutation strategies are automatically adopted according to their performance during the optimization. This also sets the user free from choosing an appropriate set of control parameters for each optimization problem. However, the use of four control parameters in the unified mutation strategy might not be optimal. In this study, we will systematically tune the number of control parameters in the unified mutation strategy using 14 unimodal and multimodal functions from the CEC2005 [21].

The rest of the paper is organized as follows: in Section 2, the standard differential evolution algorithm and its multiple mutation strategies are reviewed. In Section 3, the adaptive unified differential evolution algorithm is discussed. In Section 4, numerical tuning of control parameters in the unified mutation strategy and benchmark against several conventional mutation strategies are presented. Conclusions are provided in Section 5.

II. Standard differential evolution algorithm

The differential evolution algorithm starts with a population initialization. A set of NP solutions in the control parameter

978-1-5090-0623-6/16/$31.00

space is randomly generated to form the initial population. This initial population can be generated by sampling from a uniform distribution within the allowed parameter space if no prior information about the optimal solution is available, or by sampling from a known distribution (e.g., Gaussian) if some prior information is available.

After initialization, the differential evolution algorithm updates the population from one generation to the next generation until reaching a convergence condition or until the maximum number of function evaluations is reached. At each generation, the update step consists of three operations: mutation, crossover, and selection. The mutation and the crossover operations produce new candidates for the next generation population and the selection operation is used to select from among these candidates the appropriate solutions to be included in the next generation.

A. Mutation Strategies

During the mutation operation stage, for each population member (target vector) $\vec{x}_i,\ i = 1, 2, 3, \cdots, NP$ at generation G, a new mutant vector $\vec{v}_i$ is generated by following a mutation strategy. Some commonly used conventional mutation strategies are [5], [6], [11]:

$$\text{DE/rand/1}: \ \vec{v}_i = \vec{x}_{r_1} + F_{xc}(\vec{x}_{r_2} - \vec{x}_{r_3}) \tag{1}$$

$$\text{DE/rand/2}: \ \vec{v}_i = \vec{x}_{r_1} + F_{xc}(\vec{x}_{r_2} - \vec{x}_{r_3}) + F_{xc}(\vec{x}_{r_4} - \vec{x}_{r_5}) \tag{2}$$

$$\text{DE/best/1}: \ \vec{v}_i = \vec{x}_b + F_{xc}(\vec{x}_{r_1} - \vec{x}_{r_2}) \tag{3}$$

$$\text{DE/best/2}: \ \vec{v}_i = \vec{x}_b + F_{xc}(\vec{x}_{r_1} - \vec{x}_{r_2}) + F_{xc}(\vec{x}_{r_3} - \vec{x}_{r_4}) \tag{4}$$

$$\text{DE/current-to-best/1}: \ \vec{v}_i = \vec{x}_i + F_{cr}(\vec{x}_b - \vec{x}_i) + F_{xc}(\vec{x}_{r_1} - \vec{x}_{r_2}) \tag{5}$$

$$\text{DE/current-to-best/2}: \ \vec{v}_i = \vec{x}_i + F_{cr}(\vec{x}_b - \vec{x}_i) + F_{xc}(\vec{x}_{r_1} - \vec{x}_{r_2}) + F_{xc}\ (\vec{x}_{r_3} - \vec{x}_{r_4}) \tag{6}$$

$$\text{DE/current-to-rand/1}: \ \vec{v}_i = \vec{x}_i + F_{cr}(\vec{x}_{r_1} - \vec{x}_i) + F_{xc}\ (\vec{x}_{r_2} - \vec{x}_{r_3}) \tag{7}$$

$$\text{DE/current-to-rand/2}: \ \vec{v}_i = \vec{x}_i + F_{cr}(\vec{x}_{r_1} - \vec{x}_i) + F_{xc}\ (\vec{x}_{r_2} - \vec{x}_{r_3}) + F_{xc}\ (\vec{x}_{r_4} - \vec{x}_{r_5}) \tag{8}$$

$$\text{DE/rand-to-best/1}: \ \vec{v}_i = \vec{x}_{r_1} + F_{cr}(\vec{x}_b - \vec{x}_i) + F_{xc}\ (\vec{x}_{r_2} - \vec{x}_{r_3}) \tag{9}$$

$$\text{DE/rand-to-best/2}: \ \vec{v}_i = \vec{x}_{r_1} + F_{cr}(\vec{x}_b - \vec{x}_i) + F_{xc}(\vec{x}_{r_2} - \vec{x}_{r_3}) + F_{xc}(\vec{x}_{r_4} - \vec{x}_{r_5}) \tag{10}$$

where the integers r_1, r_2, r_3, r_4 and r_5 are chosen randomly from the interval $[1, NP]$ and are different from the current index i, F_{xc} is a real scaling factor that controls the amplification of the differential variation, $\vec{x}_b$ is the best solution among the NP population members at the generation G, and F_{cr} is a weight for the combination between the original target vector and the best parent vector or the random parent vector. The strategy DE/rand/1 proposed in the original paper of Storn and Price is the most widely used mutation strategy. It has stronger exploration capability but may converge more slowly than the strategies that use the best solution from the parent generation. The strategy DE/rand/2 uses two difference vectors and may result in a better solution after mutation than the strategies that use one difference vector [17]. The strategies DE/best/1 and DE/best/2 take advantage of the best solution found in the parent population and have a faster convergence towards the optimal solution [22]. However, they may become stuck at a local minimum point during multimodal function optimization. The DE/current-to-best/1 and DE/current-to-best/2 strategies provide a compromise between exploitation of the best solution and exploration of the parameter space. The DE/current-to-rand/1 and DE/current-to-rand/2 mutation strategies are rotation-invariant strategies [16]. The DE/rand-to-best/ strategies are similar to the DE/current-to-best/ strategies, but larger diversity of the solutions after mutation is attained by using a randomly selected parent vector instead of the current target parent vector.

B. Crossover

A crossover operation between the new generated mutant vector $\vec{v}_i$ and the target vector $\vec{x}_i$ is used to further increase the diversity of the new candidate solution. This operation combines the two vectors into a new trial vector $\vec{U}_i, i = 1, 2, 3, \cdots, NP$, where the components of the trial vector are obtained from the components of $\vec{v}_i$ or $\vec{x}_i$ according to a crossover probability Cr. In the binomial crossover scheme for a D dimensional control parameter space, the new trial vector $\vec{U}_i,\ i = 1, 2, \cdots, NP$ is generated using the following rule:

$$\vec{U}_i = (u_{i1}, u_{i2}, \cdots, u_{iD}) \tag{11}$$

$$u_{ij} = \begin{cases} v_{ij}, & \text{if } \ \text{rand}_j \leq Cr \ \text{ or } j = \text{mbr}_i \\ x_{ij}, & \text{otherwise} \end{cases} \tag{12}$$

where rand_j is a randomly chosen real number in the interval $[0, 1]$, and the index mbr_i is a randomly chosen integer in the range $[1, D]$. This ensures that the new trial vector contains at least one component from the new mutant vector.

C. Selection

The newly generated trial solution $\vec{U}_i$ is checked against the boundary of the control parameter space. If the solution lies outside the boundary, a new trial solution is generated by randomly sampling from solutions within the boundary.

The selection operation in DE is based on a one-to-one comparison. The new trial solution $\vec{U}_i$ is checked against the original target parent solution $\vec{x}_i$. If the new trial solution produces a better objective function value, it will be put into the next generation $(G+1)$ population. Otherwise, the original parent is kept in the next generation population.

The above procedure is repeated for all NP parents to generate the next generation population. Many generations are used to attain the final globally optimal solution.

III. The Adaptive Unified Differential Evolution Algorithm

Ten different mutation strategies have been proposed for the standard differential evolution algorithm (Eqs. 1-10). While DE/rand/1/bin has been widely used, DE/best/1/bin was shown to have better performance in a number of optimization test examples [22]. The presence of multiple mutation strategies complicates the use of the differential evolution algorithm. Recently, we proposed a single mutation expression that can unify most conventional mutation strategies used by the differential evolution algorithm. This single unified mutation expression can be written as:

$$\begin{aligned} \vec{v}_i \;=\; & \vec{x}_i + F_1(\vec{x}_b - \vec{x}_i) + F_2(\vec{x}_{r_1} - \vec{x}_i) \\ & + F_3(\vec{x}_{r_2} - \vec{x}_{r_3}) + F_4(\vec{x}_{r_4} - \vec{x}_{r_5}) \end{aligned} \tag{13}$$

Here, the second term on the right-hand side of equation (13) denotes the contribution from the best solution found in the current generation, the third term denotes the rotationally invariant contribution from the random solution [16], and the fourth and fifth terms are the same terms as those used in the original differential evolution algorithm to account for the contribution from the difference of parent solutions. Those last three terms divert the mutated solution away from the best solution and help to improve the algorithm's exploration of the decision parameter space. The four parameters F_1, F_2, F_3 and F_4 are the weights from each contribution. This unified expression represents a combination of exploitation (using the best found solution) and exploration (using randomly chosen solutions) when generating the new mutant solution.

From the above equation, one can see that for $F_1 = 0$, $F_2 = 1$, and $F_4 = 0$, this equation reduces to DE/rand/1; for $F_1 = 0$, $F_2 = 1$, and $F_3 = F_4$, it reduces to DE/rand/2; for $F_1 = 1$, $F_2 = 0$, and $F_4 = 0$, it reduces to DE/best/1; for $F_1 = 1$, $F_2 = 0$, and $F_3 = F_4$, it reduces to DE/best/2; for $F_2 = 0$ and $F_4 = 0$, it reduces to DE/current-to-best/1; for $F_2 = 0$ and $F_3 = F_4$, it reduces to DE/current-to-best/2; for $F_1 = 0$, and $F_4 = 0$, it reduces to DE/current-to-rand/1; for $F_1 = 0$, and $F_3 = F_4$, it reduces to DE/current-to-rand/2; for $F_2 = 1$, and $F_4 = 0$, it reduces to DE/rand-to-best/1; for $F_2 = 1$, and $F_3 = F_4$, it reduces to DE/rand-to-best/2. Using the single equation (13), the ten mutation strategies of the standard differential evolution algorithm can be written as a single expression. This new expression provides an opportunity to explore more broadly the space of mutation operators. Using a different set of parameters F_1, F_2, F_3, F_4, a new mutation strategy can be achieved. By expanding the space of mutation strategies (using the unified mutation strategy), it is possible to find a better optimized solution than the conventional standard differential evolution algorithm in some applications. Moreover, by adaptively adjusting these parameters during the optimization process, multiple mutation strategies and their combinations can be used during different stages of optimization. Thus, the unified mutation expression has the virtue of mathematical simplicity and also provides the user with flexibility for broader exploration of different mutation strategies.

The unified mutation strategy provides a method to combine and use different mutation strategies. However, choosing appropriate control parameters F_1, F_2, F_3, F_4 by trial-and-error approach can be challenging and time consuming for application users. Also, using a set of fixed control parameters does not necessarily lead to the best performance of the algorithm since different mutation strategies might have superior performance at different points during the process of evolutionary optimization. A self-adaptive method to select these control parameters will free the user from such a burden and also improve the performance of the algorithm.

Parameter tuning has been widely used in evolutionary optimization [23], [24], [25]. In general, these methods can be classified as deterministic parameter control, adaptive parameter control, and self-adaptive parameter control [26]. In deterministic parameter control, the parameters used in the algorithm evolve following a pre-determined rule (which can be time-dependent). In adaptive parameter control, the parameters are dynamically updated based on learning during the evolution. In self-adaptive parameter control, the parameters are encoded within each individual solution and evolve together with the solution during the process of optimization. In this study, we follow a self-adaptive method based on reference [14] to allow the five control parameters (F_1, F_2, F_3, F_4 and Cr) to evolve dynamically in the unified differential evolution algorithm. This self-adaptive scheme is simple to implement and achieved good performance in a number of benchmark tests.

During the mutation stage, the self-adaptive method used in this study assumes that at generation G, each individual solution $\vec{x}_i^G$, $i = 1, 2, 3, \cdots, NP$ has a set of control parameters $F_{1,i}^G, F_{2,i}^G, F_{3,i}^G, F_{4,i}^G$ and Cr_i^G associated with it. Before generating a new mutant solution using the unified differential evolution expression (13), a new set of control parameters $F_{1,i}^{G+1}, F_{2,i}^{G+1}, F_{3,i}^{G+1}, F_{4,i}^{G+1}$ and Cr_i^{G+1} are calculated as:

$$F_{j,i}^{G+1} = \begin{cases} F_{jmin} + r_{j1}(F_{jmax} - F_{jmin}), & \text{if } \; r_{j2} < \tau_j \\ F_{j,i}^G, & \text{otherwise} \end{cases} \tag{14}$$

$$Cr_i^{G+1} = \begin{cases} Cr_{min} + r_3(Cr_{max} - Cr_{min}), & \text{if } \; r_4 < \tau_5 \\ Cr_i^G, & \text{otherwise} \end{cases} \tag{15}$$

where $r_{j1}, r_{j2}, j = 1, 2, 3, 4$, r_3, r_4 are uniform random values in the interval $[0, 1]$, F_{jmin} and F_{jmax} for $j = 1, 2, 3, 4$ are the minimum and the maximum allowed values of those control parameters, Cr_{min} and Cr_{max} are the minimum and the maximum cross-over probability, and $\tau_j, j = 1, 2, 3, 4, 5$ represents the probability to use a new value or to keep the old value for the j^{th} control parameter. The values of τ_j are normally kept small so that better control parameters associated with surviving solutions will be reused to generate the new trial solution. In this study, we set $\tau_j = 0.1$ following reference [14]. We also did numerical tests with $\tau_j = 0.05, 0.15$, and 0.2 using

Pseudo-code for the AuDE Algorithm

Step 1: Generate a set of initial control parameters (F_1, F_2, F_3, F_4, Cr) from random uniform sampling in the interval [0,1]. Generate a set of initial population members by randomly sampling NP points within the feasible control parameter space $\vec{x}$ and evaluate their objective function values $f(\vec{x})$. Set the generation number $G = 0$.

Step 2: While the stopping criterion is not satisfied, Do:

For i = 1 to NP (for each target parent solution $\vec{x}_i$):

Step 2.1: Mutation

Find a set of control parameters (for j=1,2,3,4):

$$F_{j,i}^{G+1} = \begin{cases} F_{jmin} + r_{j1}(F_{jmax} - F_{jmin}), & \text{if } \; r_{j2} < \tau_j \\ F_{1,i}^{G}, & \text{otherwise} \end{cases}$$

$$Cr_i^{G+1} = \begin{cases} Cr_{min} + r_3(Cr_{max} - Cr_{min}), & \text{if } \; r_4 < \tau_5 \\ Cr_i^{G}, & \text{otherwise} \end{cases}$$

Find a mutant solution vector using the uDE mutation strategy:

Step 2.2: Crossover

Generate a new trial solution $\vec{U}_i(u_{i1}, u_{i2}, \cdots, u_{iD})$ through a binominal crossover scheme:

$u_{ij} = v_{ij}$ if $\text{rand}_{ij}[0,1] \le Cr_i^{G+1}$ *or* $j = j_{\text{rand}}$,

otherwise $u_{ij} = x_{ij}$.

Step 2.3: Selection

Evaluate the objective function of the trial solution $f(\vec{U}_i)$.

If $f(\vec{U}_i) \le f(\vec{x}_i)$, then $\vec{x}_{i,G+1} = \vec{U}_i$,

else $\vec{x}_{i,G+1} = \vec{x}_{i,G}$.

End For

$G = G + 1$

End While

the benchmark functions in the following section and did not see significant differences for most functions. The values of F_{jmin} and F_{jmax} are set to 0 and 1 respectively in this paper. We also set $Cr_{min} = 0$ and $Cr_{max} = 1$. The selection of these values is based on the consideration that the various conventional differential evolution mutation strategies of Eq. (1-10) can be covered by the settings of those parameters, and in the literature, F_3 and F_4 are rarely greater than one. The new set of control parameters (Eqs. 14-15) are used to generate the mutant solution in Eq. 13. The initial values of these control parameters are uniform random values between the minimum and the maximum values.

IV. Numerical tuning and benchmark

A. Tuning number of control parameters

In the unified mutation strategy Eq. 13, there are three terms with control parameters F_{2-4} that represent diverting from the best solution in each generation. Those terms help improve the diversity of the trial solution and the exploration of the algorithm. However, a broader exploration might result in a slower convergence speed in search for the optimal solution in some applications. By tuning the number of control parameters used in the unified mutation strategy, one may avoid some diverting terms and improve the speed of convergence to the final optimal solution. In this study, we start with the four control parameters F_{1-4} in the original unified mutation strategy (this algorithm is called AuDE4), and gradually reduce the number of control parameters to three F_{1-3}, to two F_{1-2}, and to one F_1 by dropping the terms associated with those control parameters in the mutation strategy. This results in four algorithms AuDE4, AuDE3, AuDE2, and AuDE1 with increasing level of exploitation as the number of diverting terms decreases.

To test the performance of the four choices of the control parameters, we used 14 shifted rotated unimodal and multimodal functions from the CEC2005 test functions [21]. Here, the first function F_1 is a shifted sphere function; the second function F_2 is a shifted Schwefel's problem 1.2; the third function F_3 is a shifted rotated high conditioned elliptic function; the fourth function F_4 is a shifted Schwefel's problem 1.2 with noise; and the fifth function F_5 is a Schwefel's problem 2.6 with global optimum on bounds. Except the first function, all the other four functions are non-separable functions in the parameter search space. Those five test functions are all unimodal functions with a single minimum solution. The rest nine test functions are multimodal functions with a number of local minimum solutions. The sixth function F_6 is a shifted Rosenbrock function; the seventh function F_7 is a shifted rotated Griewank function without bounds; the eighth function F_8 is a shifted rotated Ackley function with global optimum on bounds; the ninth function F_9 is a shifted Rastrigin function; the tenth function F_{10} is a shifted rotated Rastrigin function; the eleventh function F_{11} is a shifted rotated Weierstrass function; the twelfth function F_{12} is a Schwefel problem 2.13; the thirteenth function F_{13} is a shifted expanded Griewank function plus a Rosenbrock function; and the fourteenth function F_{14} is a shifted rotated expanded Scaffer F6 function. Except the ninth function, all the other multimodal functions are non-separable in the search space.

Using the above 14 test functions, we carried out numerical optimization with dimensions $N = 10$, 30, and 50 respectively in the control variable search space. The maximum number of function evaluations is set to $10,000N$. The population size (NP) for the 10, 30, and 50 dimensional problems is set as 50, 60, and 100, respectively. Each optimization is performed for 25 different random seeds. The average error of the objective function value and its standard deviation at the end of the maximum number of function evaluations is reported in Table I for each of the 10 dimensional objective functions, in Table II for the 30 dimensional functions, and in Table III for the 50 dimensional functions. The minimum average error of the objective value for each problem is shown in bold font. If two methods have same final solution values, we check the evolution of the solution and put the one with faster convergence in bold font. It is seen that among the 14 test functions, the AuDE3 using three control parameters F_{1-3} in the unified mutation strategy shows a better performance than the other three choices of the control parameters. In the 10 dimensional case (Table I), the AuDE3 wins 10 out of 14 test functions, while the AuDE4 wins 3 out of 14 test functions. In the 30 dimensional case (Table II), the AuDE3 wins 7 out of 14 test functions, while the AuDE4 wins 4 out of 14 test functions. In the 50 dimensional case (Table III), the AuDE3 wins 9 out of 14 test functions, while the AuDE4 wins 3 out of 14 test functions. This is probably due to a better balance of exploitation and exploration in the AuDE3 with one less diverting term than the original AuDE4 scheme. Figures 1 and 2 shows the evolution of the average error in the objective value for those test functions with 50 control variables (N=50). It is seen that the errors from both the AuDE1 and the AuDE2 decrease slowly as the number of generations increases and pre-converge to a wrong solution. This might be due to the fact that these schemes use too much information from the current solution and the best solution and hence lack exploration of the search space. On the other hand, the AuDE3 with an extra diverting term improves the exploration and achieves the best performance among those four schemes.

B. Benchmark with conventional DEs

We also compare the AuDE3 algorithm with two widely used conventional DE algorithms, DE/rand/1/bin ($F = 0.9$, $Cr = 0.9$) [6], DE/rand/1/bin ($F = 0.5$, $Cr = 0.9$) [14], [15], [22], and DE/best/1/bin ($F = 0.6$, $Cr = 0.3$) [22], and an adaptive conventional differential evolution algorithm (jDE) [14]. The numerical parameter settings used in these tests are the same as those used in the above section. The average error of the objective function value and its standard deviation at the end of the maximum number of function evaluations is reported in Table IV for each of the 10 dimensional objective functions, in Table V for the 30 dimensional functions, and in Table VI for the 50 dimensional functions. The adaptive unified differential evolution algorithm (AuDE3) in this study shows very good performance in these tests. Its performance is superior to the other algorithms shown in 9, 11, and 10 out of the total 14 test examples in 10, 30 and 50 dimensions, respectively.

V. Conclusions

In this study, we systematically tuned a recently proposed adaptive unified differential evolution algorithm for global optimization. In comparison to the standard differential evolution algorithm that normally contains multiple mutation strategies, this unified method has the advantages of both mathematical simplicity and flexibility for exploring the space of mutation operators. However, this algorithm involves more control parameters than the conventional DE does. Instead of three control parameters, F_{cr}, F_{xc} and Cr, as in the conventional DE, the unified DE has five control parameters, F_1, F_2, F_3, F_4 and Cr. Even with self adaptive parameter control, the use of three diverting terms in the mutation strategy may still slow down the speed of convergence to an optimal solution. Using 14 numerical test functions from CEC2005, we found that the unified mutation strategy with two diverting terms F_{1-3} improves the convergence speed over the original unified mutation strategy with three diverting terms F_{1-4} due to a better balance of exploitation and exploration. We also compared the performance of the tuned scheme with three control parameters F_{1-3} with the performance of several conventional DEs. We found that the tuned scheme scales well with 10, 30, and 50 dimensional test problems and is superior to those conventional methods. In future study, we will further benchmark the tuned adaptive unified DE with some more advanced DEs proposed in recent studies [15], [17], [18], [19]. We will also continue to explore different adaptive methods to further improve the performance of the unified differential evolution algorithm.

Acknowledgment

The work was supported by the U.S. Department of Energy under Contract No. DE-AC02-05CH11231. This research used computer resources at the National Energy Research Scientific Computing Center and at the National Center for Computational Sciences.

References

[1] D. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning. Reading, MA, USA: Addison-Wesley, 1989.

[2] I. Rechenberg, Evolutions Strategie: Optimierung Technischer Systeme Nach Prinzipien der Biologischen Evolution. Stuttgart, Germany: Fromman-Holzboog, 1973.

[3] L. Fogel, J. Owens, and M. Walsh, Artificial Intelligence Through Simulated Evolution. New York, NY, USA: Wiley, 1966.

[4] Kennedy, J.; Eberhart, R. (1995). "Particle Swarm Optimization". Proceedings of IEEE International Conference on Neural Networks IV. pp. 1942-1948. doi:10.1109/ICNN.1995.488968.

[5] R. Storn and K. Price, "Differential evolution: A simple and efficient adaptive scheme for global optimization over continuous spaces," ICSI, USA, Tech. Rep. TR-95-012, 1995.

[6] R. Storn and K. Price, "Differential evolution - A simple and efficient heuristic for global optimization over continuous spaces," Journal of Global Optimization 1a1:341-359, (1997).

[7] K. Price, R. Storn, and J. Lampinen, *Differential Evolution - A Practical Approach to Global Optimization*, Berlin, Germany: Springer, 2005.

[8] U. K. Chakraborty (ed.), *Advances in Differential Evolution*, Springer, Berlin, 2008.

[9] A. Qing, *Differential Evolution: Fundamentals and Applications in Electrical Engineering*, John Wiley, New York, 2009.

[10] J. Zhang and A. C. Sanderson, *Adaptive Differential Evolution: A Robust Approach to Multimodal Problem Optimization*, Springer, Berlin, Germany, 2009.

[11] F. Neri and V. Tirronen, "Recent advances in differential evolution: a survey and experimental analysis," Artif. Intell. Rev. 33, p. 61, 2010.

[12] S. Das and P. N. Suganthan, "Differential evolution: a survey of the state-of-the-art," IEEE Transactions on Evolutionary Computation, vol. 15, no. 1, pp. 4-31, 2011.

[13] M. M. Ali and A. Torn, "Population set based global optimization algorithms: Some modifications and numerical studies," Comput. Oper. Res., vol. 31, no. 10, pp. 1703-1725, 2004.

[14] J. Brest, S. Greiner, B. Bovskovic, M. Mernik, and V. Zumer, "Self adapting control parameters in differential evolution: A comparative study on numerical benchmark problems," IEEE Transactions on Evolutionary Computation, vol. 10, no. 6, pp. 646-657, 2006.

[15] J. Zhang and A. C. Sanderson, "JADE: adaptive differential evolution with optional external archive," IEEE Transactions on Evolutionary Computation, vol. 13, no. 5, pp. 945-958, 2009.

[16] K. V. Price, "An introduction to differential evolution," in New Ideas in Optimization, D. Corne, M. Dorigo, and V. Glover, Eds. London, U.K.: McGraw-Hill, 1999, pp. 79-108.

[17] 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, pp. 398-417, 2009.

[18] R. Mallipeddi, P. N. Suganthan, Q. K. Pan, M. F. Tasgetiren, "Differential evolution algorithm with ensemble of parameters and mutation strategies," Applied Soft Computing 11, p. 1679, 2011.

[19] Y. Wang, Z. Cai, Q. Zhang, "Differential evolution with composite trial vector generation strategies and control parameters," IEEE transcations on Evolutionary Computation, vol. 15, no.1, p. 55, 2011.

[20] J. Qiang and C. Mitchell, "An Adaptive Unified Differential Evolution Algorithm for Global Optimization," Lawrence Berkeley National Laboratory report lbnl-6853e, 2014.

TABLE I
PERFORMANCE COMPARISON OF THE UNIFIED MUTATION STRATEGY WITH DIFFERENT NUMBER OF CONTROL PARAMETERS FOR $N = 10$.

Function	AuDE1		AuDE2		AuDE3		AuDE4	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std
F_1	2.0338969E+03	9.3721533E+02	5.6325899E-03	2.2555418E-02	**0.0000000E+00**	0.0000000E+00	0.0000000E+00	0.0000000E+00
F_2	5.0996902E+03	2.2063905E+03	4.1434819E+02	3.7722535E+02	**0.0000000E+00**	0.0000000E+00	0.0000000E+00	0.0000000E+00
F_3	2.1763933E+07	1.5157812E+07	5.2378944E+05	6.1785117E+05	2.3677544E+02	5.1772101E+02	**6.8763106E-03**	8.6763986E-03
F_4	5.8729227E+03	2.9253368E+03	7.0903529E+02	6.6746937E+02	**0.0000000E+00**	0.0000000E+00	0.0000000E+00	0.0000000E+00
F_5	9.7702218E+03	1.9706674E+03	2.2192744E+03	8.3893501E+02	**0.0000000E+00**	0.0000000E+00	2.8774139E-11	1.9151445E-11
F_6	9.7167416E+07	1.2161747E+08	2.1262685E+02	4.2812623E+02	4.7838949E-01	1.2954849E+00	**1.6520062E-10**	8.0925642E-10
F_7	8.1923145E+01	6.0474423E+01	2.6000284E+00	2.6148173E+00	**8.0046650E-02**	4.7315555E-02	1.5160684E-01	5.0155327E-02
F_8	2.0306339E+01	5.4210527E-02	2.0330802E+01	8.7225643E-02	2.0348756E+01	6.9117752E-02	2.0365495E+01	8.4086814E-02
F_9	1.4683162E+01	5.4690602E+00	5.6911786E+00	3.2760469E+00	**0.0000000E+00**	0.0000000E+00	0.0000000E+00	0.0000000E+00
F_{10}	5.0389950E+01	1.6130435E+01	1.8030726E+01	8.6441699E+00	**6.2387856E+00**	2.5361897E+00	1.0329532E+01	2.7452906E+00
F_{11}	7.5674532E+00	1.0150846E+00	5.2437953E+00	9.2782173E-01	**2.1324491E+00**	1.4317264E+00	6.0780595E+00	6.6367522E-01
F_{12}	1.3920204E+04	6.8740118E+03	1.7328730E+03	1.9587730E+03	4.4314780E+02	6.5549769E+02	**8.7798535E+01**	2.9202534E+02
F_{13}	1.5594793E+00	7.3668280E-01	5.9615231E-01	2.1632779E-01	**5.3060953E-01**	9.9485625E-02	6.1782013E-01	1.1162170E-01
F_{14}	3.3039004E+00	3.1349699E-01	3.0411386E+00	3.9917555E-01	**2.3760892E+00**	3.5611190E-01	3.2144005E+00	1.5872720E-01

TABLE II
PERFORMANCE COMPARISON OF THE UNIFIED MUTATION STRATEGY WITH DIFFERENT NUMBER OF CONTROL PARAMETERS FOR $N = 30$.

Function	AuDE1		AuDE2		AuDE3		AuDE4	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std
F_1	2.3199005E+04	6.5761772E+03	3.2929972E+01	5.4135337E+01	**0.0000000E+00**	0.0000000E+00	0.0000000E+00	0.0000000E+00
F_2	3.1831924E+04	7.5106824E+03	1.2738978E+04	3.2397784E+03	**7.9580786E-14**	3.9382276E-14	7.5387493E-09	9.8461919E-09
F_3	2.8646671E+08	1.3774951E+08	4.3063112E+07	1.8055405E+07	**1.4104564E+05**	5.6480228E+04	3.3328494E+05	1.7528356E+05
F_4	3.8593489E+04	7.2205851E+03	1.8106322E+04	4.1743825E+03	9.8061342E+00	3.0106320E+01	**2.9240981E-03**	3.5283738E-03
F_5	2.4199749E+04	2.4766656E+03	1.2814950E+04	1.9197015E+03	2.2921628E+03	5.1893203E+02	**5.2281598E+02**	2.7823219E+02
F_6	5.6284383E+09	2.2012595E+09	7.9433172E+06	1.2690141E+07	1.9135795E+00	1.9917166E+00	**3.1892994E-01**	1.0815439E+00
F_7	1.0934558E+03	3.0816035E+02	9.9428190E+01	4.7903193E+01	1.5553466E-02	1.6031365E-02	**4.2331593E-03**	7.5743628E-03
F_8	2.0884124E+01	6.6730425E-02	2.0899377E+01	5.7202178E-02	2.0925805E+01	4.1211114E-02	2.0933663E+01	4.4922644E-02
F_9	1.6659194E+02	2.6506440E+01	9.1433959E+01	2.0765190E+01	**6.0493503E+00**	4.3680387E+00	2.0658434E+01	2.5445674E+00
F_{10}	3.8276231E+02	5.5565106E+01	1.6602307E+02	2.9996955E+01	**4.9181750E+01**	1.0619879E+01	1.0308975E+02	1.2476072E+01
F_{11}	3.4832391E+01	1.9920072E+00	2.9774765E+01	2.9494393E+00	**1.5619760E+01**	3.2688980E+00	3.1950896E+01	1.8354613E+00
F_{12}	4.4090360E+05	1.6254552E+05	9.4854291E+04	4.0744832E+04	2.4872612E+03	3.3555087E+03	2.2541828E+03	3.0774017E+03
F_{13}	1.5740612E+01	3.3612225E+00	6.1406720E+00	1.7179626E+00	**3.7508460E+00**	3.6730826E-01	4.3259334E+00	5.1912443E-01
F_{14}	1.2447443E+01	4.0248807E-01	**1.1926446E+01**	3.9935099E-01	1.2519872E+01	3.3397551E-01	1.3027686E+01	1.6152738E-01

TABLE III
PERFORMANCE COMPARISON OF THE UNIFIED MUTATION STRATEGY WITH DIFFERENT NUMBER OF CONTROL PARAMETERS FOR $N = 50$.

Function	AuDE1		AuDE2		AuDE3		AuDE4	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std
F_1	5.1856589E+04	6.7974550E+03	2.8246219E+02	5.5217774E+02	**0.0000000E+00**	0.0000000E+00	0.0000000E+00	0.0000000E+00
F_2	7.5187338E+04	1.1919725E+04	2.7766157E+04	5.4695104E+03	**1.7958065E-04**	2.5805315E-04	4.5671997E-02	2.1860679E-02
F_3	7.5755255E+08	2.5458235E+08	1.0210640E+08	2.6879967E+07	**5.0050682E+05**	1.9314595E+05	8.3315453E+05	2.6375077E+05
F_4	8.9950676E+04	1.9983361E+04	3.9312211E+04	6.7685914E+03	1.0955475E+03	8.7681772E+02	**2.3191438E+02**	1.4042746E+02
F_5	3.0398831E+04	2.1384404E+03	2.2562995E+04	1.9251269E+03	5.2945908E+03	9.3906237E+02	**3.0930583E+03**	4.1191004E+02
F_6	1.3634850E+10	4.4247397E+09	3.6818421E+07	7.3987899E+07	**2.8137956E+00**	2.0888036E+00	1.3828013E+01	1.1792919E+01
F_7	2.2143471E+03	3.2836830E+02	2.3587952E+02	6.7977979E+01	**5.6083296E-03**	9.6948211E-03	5.5135052E-03	8.2366409E-03
F_8	2.1021765E+01	8.4682493E-02	2.1118551E+01	4.4534042E-02	2.1122542E+01	3.2206483E-02	2.1139046E+01	2.0175323E-02
F_9	3.7926705E+02	3.4655648E+01	2.0246597E+02	2.9495124E+01	**4.2226030E+01**	1.7862920E+01	8.3467911E+01	7.4145583E+00
F_{10}	7.7851850E+02	6.1639840E+01	4.2413641E+02	5.5478479E+01	**1.1441995E+02**	2.2162231E+01	2.4785630E+02	1.9896502E+01
F_{11}	6.4940431E+01	2.5576048E+00	5.7695554E+01	4.0837813E+00	**3.9944311E+01**	1.0455411E+01	6.2506353E+01	1.6870301E+00
F_{12}	2.1132313E+06	4.0917483E+05	3.7767888E+05	7.4453808E+04	9.4551459E+03	7.9832132E+03	**7.2963078E+03**	4.6992423E+03
F_{13}	3.9000077E+01	7.1356006E+00	1.4413743E+01	3.6040257E+00	**1.0119648E+01**	7.5038179E-01	1.1694055E+01	6.2544958E-01
F_{14}	2.1763047E+01	4.5878525E-01	**2.1135427E+01**	5.7351314E-01	2.2254733E+01	3.8187439E-01	2.2686872E+01	2.7431866E-01

TABLE IV
PERFORMANCE COMPARISON OF DIFFERENT DE STRATEGIES FOR $N = 10$.

Function	rand/1/bin (0.9,0.9)		rand/1/bin (0.5,0.9)		best/1/bin (0.6,0.3)		jDE		AuDE3	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
F_1	0.0000000E+00	0.0000000E+00	1.1600453E-05	6.7763915E-06	**0.0000000E+00**	0.0000000E+00	2.2737368E-15	1.1138990E-14	0.0000000E+00	0.0000000E+00
F_2	0.0000000E+00	0.0000000E+00	5.2947817E-02	3.1812454E-02	0.0000000E+00	0.0000000E+00	8.2032011E-09	5.6288682E-09	**0.0000000E+00**	0.0000000E+00
F_3	**2.7436183E-03**	1.9246452E-03	2.7960187E+04	1.0797875E+04	1.9969938E+05	8.8750793E+04	1.8085329E+03	6.9729592E+02	2.3677544E+02	5.1772101E+02
F_4	6.5907689E-08	1.1377208E-07	8.0278412E-01	3.2090327E-01	0.0000000E+00	0.0000000E+00	4.1926821E-03	2.6378655E-03	**0.0000000E+00**	0.0000000E+00
F_5	3.0234581E+01	7.0282410E+00	7.5117681E+02	1.4761628E+02	**0.0000000E+00**	0.0000000E+00	1.5947931E+01	4.2887970E+00	0.0000000E+00	0.0000000E+00
F_6	5.6672991E+00	2.0288077E+00	1.2190600E+02	4.3514187E+01	2.0705489E+00	1.3535926E+00	2.0976418E+00	1.5543555E+00	**4.7838949E-01**	1.2954849E+00
F_7	5.3679379E-01	7.0823728E-02	6.3233369E-01	8.4555716E-02	1.6378389E-01	5.7131860E-02	1.4336614E-01	3.1907309E-02	**8.0046650E-02**	4.7315555E-02
F_8	2.0338342E+01	6.7732253E-02	2.0340375E+01	8.7961754E-02	2.0350417E+01	6.5733178E-02	2.0333058E+01	6.9860187E-02	2.0348756E+01	6.9117752E-02
F_9	2.4927923E+01	3.8553340E+00	2.4869593E+01	3.2228950E+00	1.4725394E+00	1.3531443E+00	2.1613194E+00	7.9532603E-01	**0.0000000E+00**	0.0000000E+00
F_{10}	3.3944239E+01	4.4434536E+00	4.1290421E+01	5.8501187E+00	1.3738418E+01	3.5896022E+00	1.7154255E+01	3.1317709E+00	**6.2387856E+00**	2.5361897E+00
F_{11}	8.1169937E+00	8.3923093E-01	8.9664471E+00	5.7881595E-01	6.2555724E+00	6.7614688E-01	4.9552116E+00	6.9448385E-01	**2.1324491E+00**	1.4317264E+00
F_{12}	6.3814388E+02	2.4652132E+02	3.3414181E+03	1.0060128E+03	**1.6773491E+02**	4.0512937E+02	1.1206846E+02	5.3707647E+01	4.4314780E+02	6.5549769E+02
F_{13}	2.6586147E+00	3.4527250E-01	3.4535876E+00	3.2945086E-01	6.0300527E-01	1.5142810E-01	7.1776035E-01	1.1885232E-01	**5.3060953E-01**	9.9485625E-02
F_{14}	3.4981003E+00	1.4757772E-01	3.7472745E+00	1.3941907E-01	3.2142822E+00	2.6952966E-01	3.3314049E+00	1.6609668E-01	**2.3760892E+00**	3.5611190E-01

TABLE V
PERFORMANCE COMPARISON OF DIFFERENT DE STRATEGIES FOR $N = 30$.

Function	rand/1/bin (0.9,0.9)		rand/1/bin (0.5,0.9)		best/1/bin (0.6,0.3)		jDE		AuDE3	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
F_1	4.0042068E-02	8.8307992E-03	1.0409654E+04	1.2104588E+03	5.4569682E-14	1.9559423E-14	5.3842086E-12	2.1210759E-12	**0.0000000E+00**	0.0000000E+00
F_2	1.5757241E+03	2.9986085E+02	3.3304339E+04	3.3040277E+03	2.8456738E+01	2.1763623E+01	1.2267751E+02	1.9971381E+01	**7.9580786E-14**	3.9382276E-14
F_3	2.4548382E+07	4.2367728E+06	2.0209261E+08	3.6717114E+07	2.8650801E+07	8.7024844E+06	4.2126081E+06	9.0229399E+05	**1.4104564E+05**	5.6480228E+04
F_4	8.9192323E+03	1.5487786E+03	4.0898995E+04	3.8311927E+03	4.4456661E+02	3.5711772E+02	5.9860186E+03	1.3586903E+03	**9.8061342E+00**	3.0106320E+01
F_5	9.4191660E+03	5.8526015E+02	1.8207668E+04	1.2946165E+03	**1.6502553E+03**	4.0466493E+02	5.1414490E+03	7.3617359E+02	2.2921628E+03	5.1893203E+02
F_6	1.5925815E+04	4.1294933E+03	8.7971860E+08	2.2523826E+08	2.9464409E+01	2.0417706E+01	3.5691109E+01	3.4591071E+00	**1.9135795E+00**	1.9917166E+00
F_7	1.0166459E+00	2.3860761E-02	5.2188218E+02	5.9920405E+01	**1.2116428E-02**	1.0649478E-02	6.2153108E-02	1.9692525E-02	1.5553466E-02	1.6031365E-02
F_8	2.0949387E+01	4.7929997E-02	2.0952678E+01	3.7664812E-02	2.0939889E+01	4.3785670E-02	2.0942710E+01	4.5859964E-02	2.0925805E+01	4.1211114E-02
F_9	2.2875580E+02	1.1624656E+01	2.7182950E+02	1.1788136E+01	9.7903961E+00	3.4544417E+00	4.3379156E+01	6.0438936E+00	**6.0493503E+00**	4.3680387E+00
F_{10}	2.5627592E+02	1.1178713E+01	3.4069565E+02	1.3533140E+01	1.6335374E+02	1.7418266E+01	1.9153300E+02	1.3487459E+01	**4.9181750E+01**	1.0619879E+01
F_{11}	3.9625818E+01	1.0999765E+00	3.9586892E+01	9.2565013E-01	3.7911211E+01	1.0955156E+00	2.7922660E+01	1.1772193E+00	**1.5619760E+01**	3.2688980E+00
F_{12}	3.9682312E+05	4.4826984E+04	6.9160653E+05	7.3840596E+04	7.8503639E+03	1.0178754E+04	3.7835219E+04	5.6300539E+03	**2.4872612E+03**	3.3555087E+03
F_{13}	2.1428267E+01	1.1650938E+00	3.9428286E+01	3.6998405E+00	7.7156768E+00	7.5339803E-01	6.0425845E+00	5.8237527E-01	**3.7508460E+00**	3.6730826E-01
F_{14}	1.3357126E+01	1.5099584E-01	1.3469751E+01	1.0284091E-01	1.3251791E+01	1.3541026E-01	1.2910809E+01	1.5619979E-01	**1.2519872E+01**	3.3397551E-01

TABLE VI
PERFORMANCE COMPARISON OF DIFFERENT DE STRATEGIES FOR $N = 50$.

Function	rand/1/bin (0.9,0.9)		rand/1/bin (0.5,0.9)		best/1/bin (0.6,0.3)		jDE		AuDE3	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
F_1	3.0927386E+03	3.3806897E+02	5.8565996E+04	3.3438702E+03	7.0485839E-14	2.4276865E-14	9.7900730E-04	1.9327500E-04	**0.0000000E+00**	0.0000000E+00
F_2	5.3784094E+04	5.5778455E+03	1.1253028E+05	8.7430565E+03	1.6005087E+04	3.0178019E+03	1.3310258E+04	1.5506144E+03	**1.7958065E-04**	2.5805315E-04
F_3	3.4245986E+08	5.7162253E+07	1.0891557E+09	1.3447534E+08	1.0730390E+08	1.9084991E+07	4.0242215E+07	8.4701652E+06	**5.0050682E+05**	1.9314595E+05
F_4	7.3229680E+04	5.7847085E+03	1.2960611E+05	7.6975439E+03	2.8131986E+04	5.1831596E+03	4.6937176E+04	4.3890970E+03	**1.0955475E+03**	8.7681772E+02
F_5	2.2193246E+04	1.4385840E+03	3.1059695E+04	1.4857210E+03	**4.0664744E+03**	7.8421179E+02	1.4083438E+04	1.0919037E+03	5.2945908E+03	9.3906237E+02
F_6	2.1235168E+08	4.0623867E+07	1.6371570E+10	2.5799614E+09	5.9170498E+01	2.5931487E+01	4.1049263E+02	5.5862000E+01	**2.8137956E+00**	2.0888036E+00
F_7	2.9435586E+02	2.3509047E+01	2.4503718E+03	1.7207500E+02	**3.5477374E-03**	5.3555807E-03	6.4442528E-01	4.5368334E-02	5.6083296E-03	9.6948211E-03
F_8	2.1142240E+01	3.0595834E-02	2.1136908E+01	3.1511796E-02	2.1136844E+01	3.2365326E-02	2.1144190E+01	4.4635915E-02	2.1122542E+01	3.2206483E-02
F_9	4.9859346E+02	1.6299490E+01	6.0849638E+02	1.5217194E+01	**3.8286541E+01**	3.7246640E+01	1.5887065E+02	1.3732574E+01	4.2226030E+01	1.7862920E+01
F_{10}	6.0219018E+02	1.9609180E+01	8.3051164E+02	3.0416330E+01	3.6701324E+02	1.7874542E+01	4.9269444E+02	4.3478100E+01	**1.1441995E+02**	2.2162231E+01
F_{11}	7.3080034E+01	1.0969651E+00	7.2906533E+01	1.6662393E+00	7.1787659E+01	1.4045202E+00	5.7315084E+01	1.7626651E+00	**3.9944311E+01**	1.0455411E+01
F_{12}	2.3216155E+06	1.7478052E+05	3.7863579E+06	3.0129766E+05	1.7232857E+04	1.1055801E+04	2.6931807E+05	3.1723774E+04	**9.4551459E+03**	7.9832132E+03
F_{13}	5.6457019E+01	3.2638241E+00	2.1127535E+02	2.8863691E+01	2.1405098E+01	1.0925048E+00	1.6828538E+01	1.1329713E+00	**1.0119648E+01**	7.5038179E-01
F_{14}	2.3082455E+01	1.6783775E-01	2.3233219E+01	1.3943413E-01	2.2943825E+01	1.3336836E-01	2.2683903E+01	1.4716163E-01	**2.2254733E+01**	3.8187439E-01

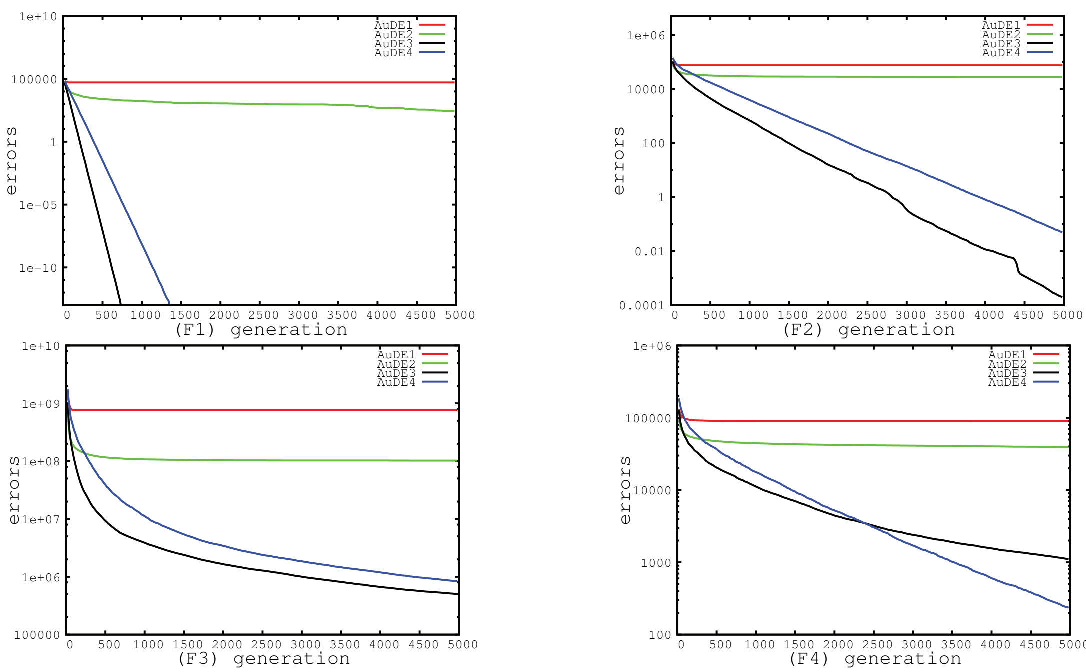

Fig. 1. Evolution of the average error in the objective value for the first four test functions with 50 control variables.

[21] P. N. Suganthan, N. Hansen, J. J. Liang, K. Deb, Y. P. Chen, A. Auger, S. Tiwari, "Problem Definitions and Evaluation Criteria for the CEC 2005 Special Session on Real-Parameter Optimization," Technical Report, Nanyang Technological University, Singapore, 2005.

[22] E. Mezura-Montes, J. Velazquez-Reyes, and C. A. Coello Coello, "A comparative study of differential evolution variants for global optimization," in Proc. Genet. Evol. Comput. Conf., 2006, pp. 485- 492.

[23] L. Davis, "Adapting operator probabilities in genetic algorithms," presented at the 3rd Int. Conf. Genetic Algorithms, Fairfax, VA, USA, 1989.

[24] A. E. Eiben, R. Hinterding, and Z. Michalewicz, "Parameter control in evolutionary algorithms," IEEE Trans. Evol. Comput., vol. 3, no. 2, pp. 124-141, 1999.

[25] J. A. Vrugt, B. A. Robinson, and J. M. Hyman, "Self-adaptive multimethod search for global optimization in real-parameter spaces," IEEE Trans. Evol. Comput., vol. 13, no. 2, pp. 243-259, Apr. 2009.

[26] A. E. Eiben and J. E. Smith, Introduction to Evolutionary Computing. Berlin, Germany: Springer, 2003.

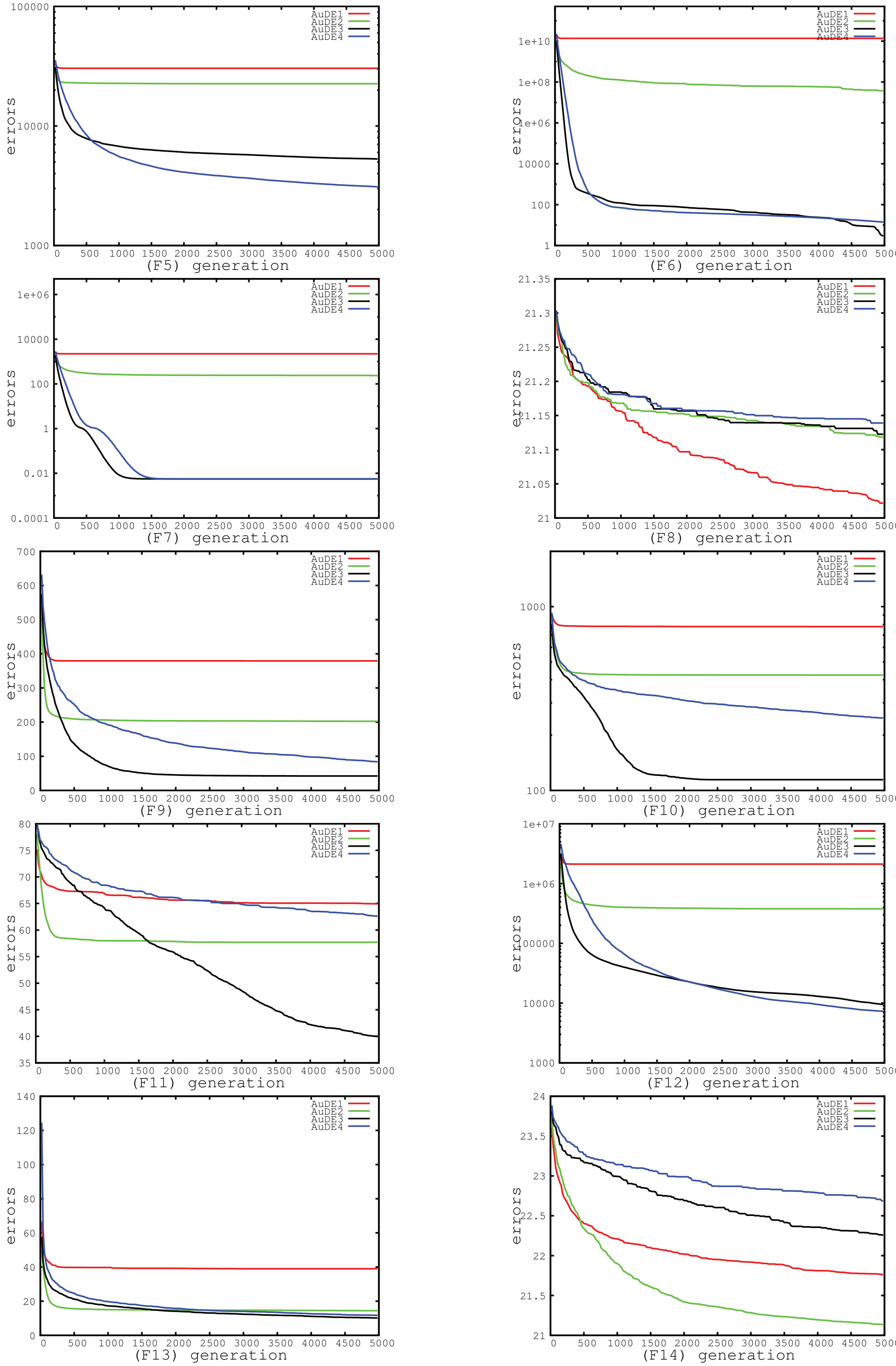

Fig. 2. Evolution of the average error in the objective value for the test function five to fourteen with 50 control variables.