International Congress on Evolutionary Methods for Design,
Optimization and Control with Applications to Industrial Problems
EUROGEN 2003
G. Bugeda, J. A- Désidéri, J. Periaux, M. Schoenauer and G. Winter (Eds)

IMPROVEMENTS TO MUTATION DONOR FORMULATION OF DIFFERENTIAL EVOLUTION

Hui-Yuan Fan[*], Jouni Lampinen[†] and George S. Dulikravich[*]

[*] MAIDO Institute, Department of Mechanical and Aerospace Engineering,
The University of Texas at Arlington, Arlington, TX 76019, USA
e-mail: {fan, dulikra}@mae.uta.edu

[†]Department of Information Technology,
Lappeenranta University of Technology, P. O. Box 20, FIN-53851 Lappeenranta, Finland
e-mail: jlampine@lut.fi

Key words: evolutionary algorithm, differential evolution, convergence, mutation operation, neural networks, aerodynamic approximation

Abstract: *As one of the most promising novel evolutionary algorithms, differential evolution has been demonstrated to be an efficient, effective and robust optimization method for nonlinear optimization. Nevertheless, the convergence rate of differential evolution is still far from ideal when it is applied to optimizing a computationally expensive objective function that is frequently encountered in engineering optimization problems. This paper proposes three new schemes to determine the donor for mutation operation in differential evolution. These modifications to the mutation operator are analyzed and compared empirically by using a suite of artificial test functions. They are further examined with two practical neural-network-based aerodynamic data approximation cases. The simulation results demonstrate that the proposed strategies are capable of accelerating the convergence rate of the differential evolution algorithms.*

1 INTRODUCTION

Evolutionary algorithms (EAs) have received a considerable of attention because of their potential in solving a wide range of difficult optimization problems. Differential evolution (DE), developed by Storn and Price [1–6], is one of the most promising evolutionary algorithms. This method has been empirically demonstrated to be an efficient and robust optimization method that outperforms some traditional EAs [7]. DE can also be easily extended for handling continuous, discrete and integer variables [8–10] and multiple non-linear and non-trivial constraints [11]. At the same time, DE is also extremely simple to implement.

Though DE has demonstrated better convergence performance than the other EAs for optimization of multi-modal functions, its convergence rate is still far from ideal. Since all EAs, including DE, work with a population of solutions rather than a single solution, many evaluations of candidate solutions are required in the optimization process. If evaluation of the candidate solutions is too time-consuming, the overall optimization time may become too long to be acceptable. In such cases, one often has no other option but to limit the algorithm to operate within an acceptable time, which may not be enough to find the global optima but enough to obtain an improved solution (that is, a sub-optimum). Thus, further study of acceleration strategies for the current DE algorithm is an important and necessary topic.

In this paper, three new schemes are proposed to determine the donor vectors for DE mutation operations. These new donor schemes form a donor vector as a convex combination of the members in a triplet of the individuals selected for mutation. The aim of these modifications is to accelerate the convergence rate of DE, thus to make it yield an acceptable solution with fewer objective function evaluations. The effectiveness of the new schemes is first assessed with artificial test functions that are widely used as performance benchmarks for EAs. Then further examination is carried out by applying DE to two neural network training cases for approximating practical aerodynamic data.

2 A SHORT OVERVIEW OF DIFFERENTIAL EVOLUTION ALGORITHM

There are several variant versions of DE [5]. In this research we use the *DE/rand/1/bin*- version, which appears to be the most frequently used variant, and is often considered as the "basic" version of DE. This particular version is briefly described as follows.

When DE is applied to solving an optimization problem formulated as $\min_X f(X)$, where X is a vector of $n \times 1$ parameters, it evolves a population of N_p candidate solutions, that is, $X_{i,G}$, $i = 1,...,N_p$, where i indexes the population and G is the generation to which the population belongs. The main operations of DE, mainly mutation, crossover and selection, are briefly discussed in turn.

The mutation operation in DE applies the vector differentials between the existing population members for determining both the degree and direction of perturbation applied to the individual that is the subject of the mutation operation. The mutation process begins by randomly selecting three individuals in the population to form a triplet. In the triplet one member is randomly taken

as the donor and the other two members are taken to make up a perturbation to the donor. The i th perturbed individual is therefore generated based on the three chosen individuals as in the form

$$V_{i,G+1} = X_{r_3,G} + F \cdot (X_{r_1,G} - X_{r_2,G}) \tag{1}$$

where $r_1, r_2, r_3 \in \{1,...,N_p\}$ are randomly selected and satisfy $r_1 \neq r_2 \neq r_3 \neq i$. The scaling constant F ($F \in [0,1+]$), introduced by Storn and Price [1] in Equation (1), is a control parameter of DE. The perturbed individual, $V_{i,G+1} = (v_{1,i,G+1},...,v_{n,i,G+1})$ and the current population member $X_{i,G+1} = (x_{1,i,G+1},...,x_{n,i,G+1})$, are then subjected to the crossover operation that finally generates the population of candidates or "trial" vectors $U_{i,G+1} = (u_{1,i,G+1},...,u_{n,i,G+1})$ as follows

$$u_{j,i,G+1} = \begin{cases} v_{j,i,G+1} & \text{if } rand_j \leq C_r \vee j = k \\ x_{j,i,G} & \text{otherwise} \end{cases} \tag{2}$$

Here, $j = 1,...,n$, $k \in \{1,...,n\}$ is the random parameter index, chosen once for every i. The crossover factor $C_r \in [0,1]$ is the other control parameter in DE that is set by the user. The population for the next generation is selected from the individuals in the current population and its corresponding trial vector according to the following rule

$$X_{i,G+1} = \begin{cases} U_{i,G+1} & \text{if } f(U_{i,G+1}) \leq f(X_{i,G}) \\ X_{i,G} & \text{otherwise} \end{cases} \tag{3}$$

3 THREE NEW DONOR SCHEMES

Three new donor schemes are proposed for the mutation operator of DE. In general formulation, these schemes can be described as a convex combination of the three members of the triplet for mutation as follows

$$donor = \sum_{i=1}^{3} \mu_i X_{r_i,G} \tag{4}$$

where the weights, $\mu_i \geq 0$, and $\sum_{i=1}^{3} \mu_i = 1$.

From this general donor definition the following special cases are derived that represent three new donor schemes investigated in the present study.

a) Arithmetic mean case

$$donor1 = \frac{1}{3} \sum_{i=1}^{3} X_{r_i,G} \tag{5}$$

b) Gauss distribution weighted mean case

$$donor2 = \sum_{i=1}^{3}(\lambda_i / \sum_{j=1}^{3}\lambda_j)X_{r_i,G}\ ,\ \lambda_j = randn_j \tag{6}$$

where $randn_j$ denotes random values that satisfy Gaussian distribution as $R(randnj) = \exp[(-(randn_j - c)^2/(2\sigma^2)]$, where we let $c = 0.5$ and $\sigma = 0.2$.

c) Uniform distribution weighted mean case

$$donor3 = \sum_{i=1}^{3}(\lambda_i / \sum_{j=1}^{3}\lambda_j)X_{r_i,G}\ ,\ \lambda_j = rand_j[0,1] \tag{7}$$

where $rand_j[0,1]$ denotes a uniformly distributed value within the range [0.0, 1.0].

These three new mutation donor schemes were defined so that each scheme determines a donor that can comprise the local information of all members in the triplet. It is expected that such use of additional local information may improve DE's performance by providing a better starting-point for the differential mutation operation and finally result in a better distribution of the trial vectors.

It should be pointed out that the original DE mutation donor scheme can be viewed as a special case of the above formulation where $\mu_3 = 1$ ($\mu_1 = \mu_2 = 0$), and hereby is denoted as

$$donor0 = X_{r_3,G} \tag{8}$$

In the search space, the triplet of individuals selected for mutation can form a trigonometric hyperplane having the three member individuals as its vertices. The donors determined by any of the three new schemes will lie (or lie with a high probability) within this trigonometric hyperplane. Their geometric interpretations are further clarified as follows.

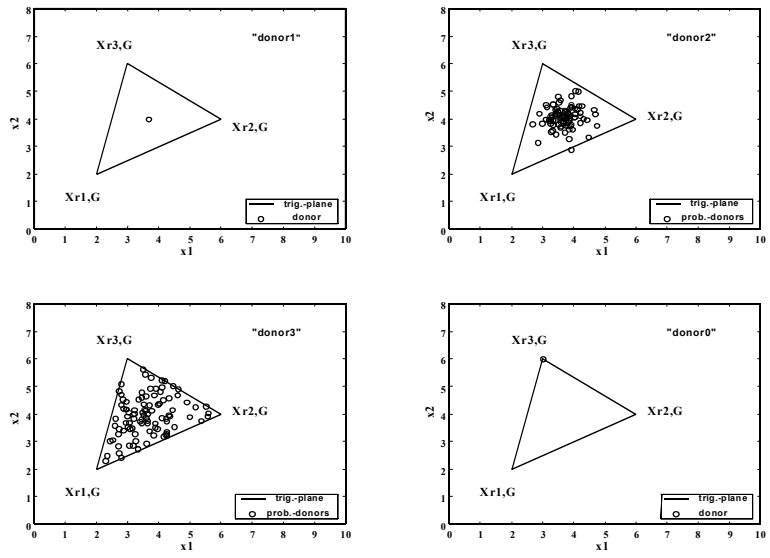

Figure 1: A geometric presentation of the mutation donor schemes with a two-dimensional optimization problem

A two-dimensional optimization problem is taken as an example for providing an easy-to-understand description. As shown in Figure 1, it is assumed that the triplet selected for mutation consists of the individuals $X_{r_1,G}$, $X_{r_2,G}$ and $X_{r_3,G}$. From the formulation (4), it can easily be deduced that the larger the weight that a member of the triplet has, the closer it "attracts" the donor to itself. For the arithmetic mean case, the new donor is simply situated at the geometric center of the trigonometric plane. The normal-distribution-weighted-mean case and the uniform-distribution-weighted-mean case determine the new donors by random process. In the former case, the probable donors are produced to statistically satisfy a normal distribution with respect to the geometric canter of the trigonometric plane. In the later case, the probable donors will be uniformly distributed over the trigonometric region. Finally, the original donor scheme in DE takes a vertex of the trigonometric plane as a donor.

4 BENCHMARK EXPERIMENTS

In this study, four functions from De Jong's test suit[12–13] and two additional functions[1,14] are chosen to empirically examine the three new mutation donor schemes for DE. These functions are listed in Table 1.

In this experimentation, DE is performed with the three newly proposed schemes, namely, $donor1$, $donor2$ and $donor3$, and the original scheme, $donor0$, to solve the test function minimization problems. All the values for the control parameters of DE are chosen based on the authors' experience and the general recommendations given in the literature[1–3,5]. These parameters are taken as $N_p = 30$, $C_r = 0.85$ and $F = 0.99$ for all the tested cases. For each test case, the experiments are repeated 100 times. All the trial runs are stopped at the 800-th generation. DE is implemented within the Matlab 5.0 system on a PC with a Pentium III processor.

The convergence histories for each test function are plotted in Figure 2. The results are averaged from 100 independent trial runs for each case. From the figure, it can be observed that for all the six function minimization problems, DE with one of the new mutation donor schemes indicates a considerably higher convergence rate than that with the original mutation donor scheme. For each function, DE with any of the three new donors also yielded a lower objective function value than that with the original donor scheme.

In the cases of functions f_1, f_2, f_4 and f_6, DE with a new donor converges to a vicinity near the global minimum at an earlier generation (for example, about 400^{th} generation), while DE with the original donor does not converge to a fairly good solution even until the specified maximum number of generations.

With functions f_3 and f_5, DE with a new mutation donor scheme also converges faster, though it does not attain a global optimum within the given maximum number of generations. Nevertheless, the solutions they obtained are much closer to their corresponding global minima than the solutions obtained by DE with the original donor.

Title	Function	n	S	X^*	$f_{\min}$
Sphere function[12,13]	$f_1(X) = \sum_{i=1}^{n} x_i^2$	30	$[-5.12,5.12]^n$	$(0,\ldots,0)^T$	0
Step function[12,13]	$f_2(X) = \sum_{i=1}^{n} \lfloor x_i + 0.5 \rfloor^2$	30	$[-5.12,5.12]^n$	$\in [-0.5,0.5]^n$	0
Ackley's function[12,13]	$f_3(X) = -20 \cdot \exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}) - \exp(\frac{1}{n}\sum_{i=1}^{n} \cos(2\pi x_i)) + 20 + e$	30	$[-20,30]^n$	$(0,\ldots,0)^T$	0
Griewangk's function[1]	$f_4(X) = \sum_{i=1}^{n} \frac{x_i^2}{4000} - \prod_{i=1}^{n} \cos(\frac{x_i}{\sqrt{i+1}}) + 1$	20	$[-400,400]^n$	$(0,\ldots,0)^T$	0
Rastrigin's function[14]	$f_5(X) = nA + \sum_{i=1}^{n} (x_i^2 - A\cos(2\pi x_i))$	20	$[-5.12,5.12]^n$	$(0,\ldots,0)^T$	0
Noisy quadratic function[12,13]	$f_6(X) = \sum_{i=1}^{n} i x_i^4 + U([0,1))$	30	$[-1.28,1.28]^n$	$(0,\ldots,0)^T$	0

Notes: n = dimensions of the functions; $f_{\min}$ = the minimum of the value of the function; X^* = the minimum; S = the feasible region; and $S \in R^n$

Table 1: Test functions

		"donor0"	"donor1"	"donor2"	"donor3"
f_1	$\overline{\Delta f}$	33.9471	0.0104	0.0077	0.0004
	$\overline{\Delta x}$	5.7567	0.0963	0.0805	0.0179
f_2	$\overline{\Delta f}$	33.8800	0.8600	0.8000	1.3200
	$\overline{\Delta x}$	5.8046	1.6927	1.6681	1.8736
f_3	$\overline{\Delta f}$	16.5471	6.3028	5.0983	1.7485
	$\overline{\Delta x}$	57.9692	25.6756	20.7955	6.6104
f_4	$\overline{\Delta f}$	10.1669	0.0628	0.0349	0.0092
	$\overline{\Delta x}$	186.0093	3.4970	2.5731	3.6102
f_5	$\overline{\Delta f}$	33.6042	10.7511	8.3715	5.9063
	$\overline{\Delta x}$	3.7444	2.3167	2.2766	2.3786
f_6	$\overline{\Delta f}$	10.0056	0.1462	0.1370	0.1099
	$\overline{\Delta x}$	1.8616	0.6407	0.6333	0.5878

Table 2: Error between the found solutions and the globally optimal solution. $\overline{\Delta f}$ refers to the average error of the objective function value, while $\overline{\Delta x}$ refers to the average Euclidean distance (within the search space) between the found point and the optimum point. The results are averaged over 100 independent trial runs.

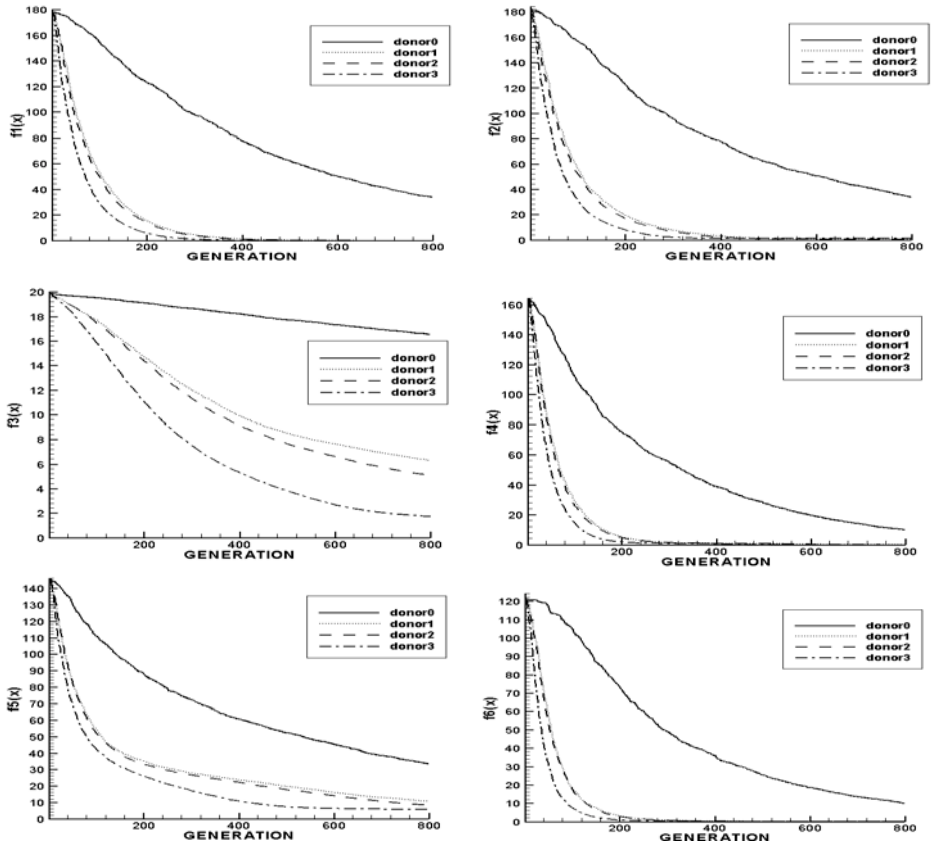

Figure 2: Convergence histories of DE with different mutation donor schemes for solving the six chosen problems. The results are averaged over 100 independent trial runs.

From Figure 2, it can also be observed that *donor*3 gives the best convergence performance for all cases. The scheme *donor*1 shows the least improvement over the standard DE. The results with the *donor*2 scheme are only slightly better than with the *donor*1 scheme. The similarity of the performances between *donor*1 and *donor*2 schemes is expected. While the *donor*1 scheme produces the donor deterministically to the center of a triangular plane, in the *donor*2 scheme the expectation for the location of the donor by Gaussian distribution is the very same point. The difference is that the *donor*2 scheme may vary the donor position around the center of the triangular plane by applying Gaussian probability distribution function.

In Table 2 some further results, also averaged from 100 independent trial runs, are presented. The value $\overline{\Delta f}_{\min} = \left| \overline{f}_{\min} - f_{\min} \right|$ represents the error between the true minimum value of the objective function and the best solution found by DE with each of the four different donors where $\overline{f}_{\min}$ is the averaged value from 100 independent trial runs. Respectively, $\overline{\Delta x} = \frac{1}{T} \sum \left\| X - X^* \right\|$ is the averaged Euclidean distance from the true optimum point to the

best-found point (defined within the search space). From the table, it can be observed that each of the three new mutation donor schemes approximates the globally optimal solutions with a higher accuracy than DE with the original donor.

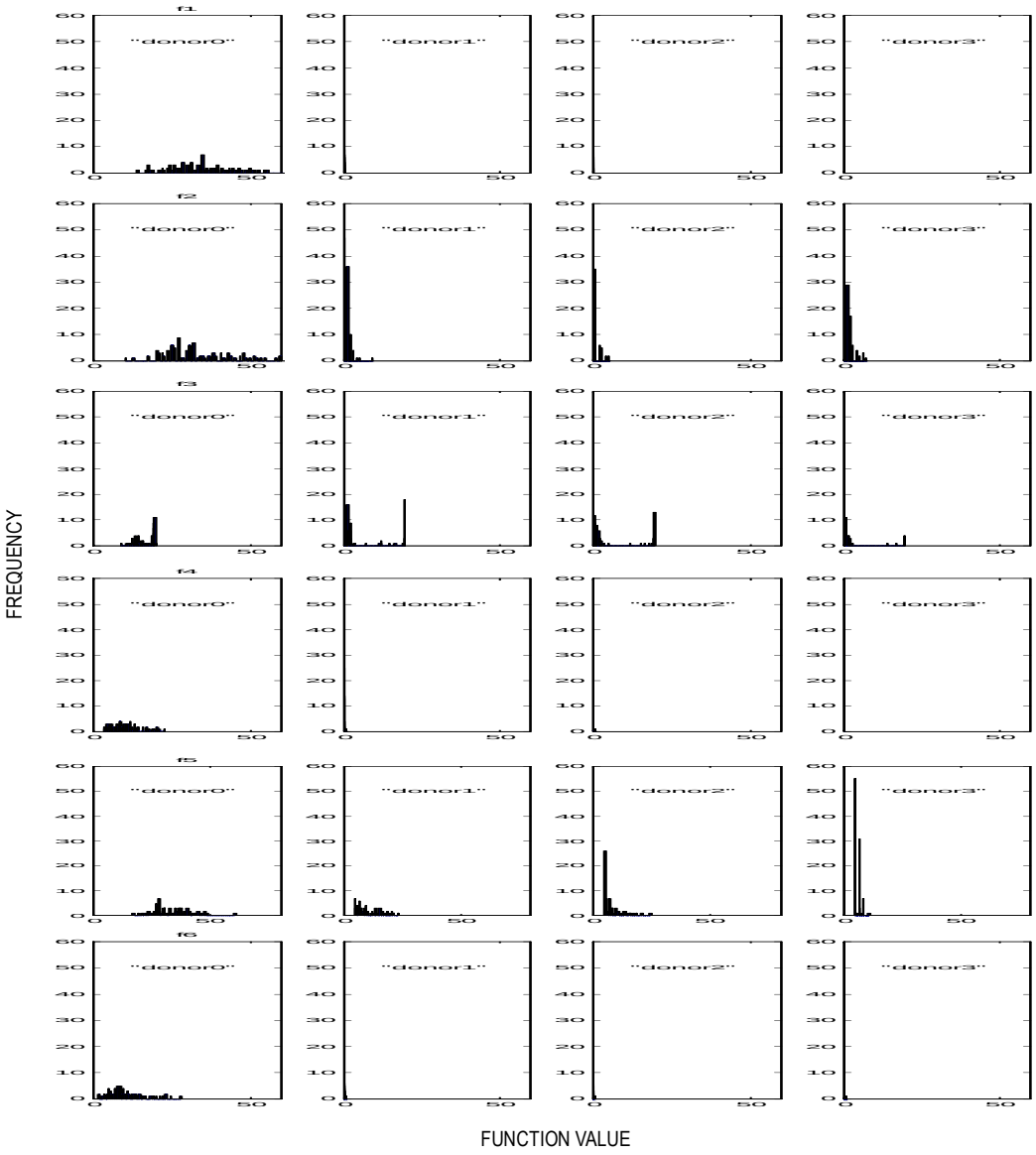

Figure 3: Histograms of the final function values of the chosen problems searched by the DE algorithm with different mutation donor schemes in the 100 independent trial runs

The convergence reliability of DE with four different mutation donor schemes is further assessed with these test functions on the basis of the 100 trial runs for each function. The results are reported in Figure 3 in the form of histograms. As shown in the figure, DE with new donors converges more frequently to the close vicinity of the global minimum for all test functions,

than DE with the original donor does. Figure 3 qualitatively shows that the three new mutation donor schemes also provide statistically higher convergence reliability than the original mutation donor scheme for the chosen functions within the given maximum number of generations.

5 VALIDATION FOR TRAINING NEURAL NETWORKS FOR AERODYNAMIC APPROXIMATIONS

The training of neural networks usually can be classified as a multi-modal nonlinear optimization problem. A wide variety of difficulties have been encountered while solving this type of problem. For example, a low convergence rate allowing only a minor convergence within the time available for computation is one of the most common practical difficulties. This is because a practical physical system is usually so complex that its approximation by a neural network may need a large network structure as well as a large number of samples to train it. Therefore, any attempts towards improving the convergence of neural network training are well motivated, since the possible progress would be very profitable for practical applications. In this section DE with the improved donors is applied to the training of multi-layered feed forward neural networks to approximate some practical aerodynamic data.

The training of a neural network can be viewed as a minimization process of the mean squared error (MSE) function of the neural network. When DE is applied to solve such an optimization problem, the individuals are no longer vectors, but weight matrices of the neural network. Due to this, the DE algorithm must evolve in synchronism a population in which the individuals are groups of weight matrices with several different sizes. So, all the DE operations are required to adapt based on such individuals. The details of implementation of DE with these individuals can be found elsewhere[15]. Accordingly, in our current application the investigated DE donors must be adapted to the matrix-group-individuals, W, as $donor = \sum_{i=1}^{3} \mu_i W_{r_i,G}$. All the other evolution parameters remain the same definitions as in the previous parts 2-4. The algorithm scheme of DE keeps its original structure.

5.1 Approximation tasks

a) Performance-prediction of a prototypic mixer

A prototypic mixer for flameless oxidation combustors was designed and studied by the lead author[16]. The mixer has the mixing chamber shaped like a "box". Its inlet channel has two equal cross-section-area parts used to input cold air and hot air to be mixed. The boundary contour between the two inlet parts is square-tooth-indented shaped, which is defined by the tooth-height h and width, w. Two parameters are selected as the criteria to describe the mixer performance. One is temperature pattern factor (TPF), and another is total pressure loss factor (TPLF). A simple performance approximation problem is to find a function mapping from the two geometric parameters, h and w, and the velocity ratio between the inlet cold and hot air streams, denoted as c, to the two performance parameters, TPF and TPLF. Mathematically, this means to approximate the function, $K = g(H)$, where $H = [h \;\; w \;\; c]^T$, $K = [TPF \;\; TPLF]^T$. The sample data were obtained from computational fluid dynamics CFD simulations by varying the

independent parameters mentioned above in certain ranges. The commercial CFD code, STAR-CD, was used for the simulations. In this example case, we select the data obtained from five h values, seven w values, and four c values. Totally, 140 input-output pairs are available that can be used to approximate the mixer performance.

b) An aerodynamic pressure five-hole probe calibration

Another example is a five-hole pressure probe with a conical tip. In the tip, one port sits at the center of the cone, and the other four ports are axisymmetrically arranged in a ring downstream. Let P_5 denote the pressure measured at the central port, and P_i, i=1,2,3,4, denote the pressures at the other four ports around the central port. The pressure coefficients are defined as: $B_i = (P_5 - P_i)/P'$, where $P' = P_5 - 0.25 \cdot \sum_{i=1}^{4} P_i$. When a multi-layered feed forward neural network is applied to fulfill the above calibration task, it can be interpreted as to train a neural network to approximate the mapping relation as $A = g(B)$, where $A = [\alpha\ \beta\ A_t\ A_s]^T$, in which α is the flow pitch angle, β is the flow yaw angle, A_t is the total pressure coefficient, A_s is the static pressure coefficient, and $B = [B_1\ B_2\ B_3\ B_4]^T$.

The probe calibration data was obtained at an air jet calibration facility. Varying the α and β in steps of 5 deg from -40 deg to +40 deg, the total number of $17 \times 17 = 289$ data points were obtained that form 289 input-output pairs, that is, $(B, A)_i$, i = 1,…289. From these input-output pairs, 153 were uniformly chosen for the following neural network training.

5.2 Experimental results and discussions

Two feed forward neural networks were trained, one for the mixer performance prediction case and another for the probe calibration case. The former had the topological structure of "3-25-2", while the latter had the topology of the type "4-20-20-4". Here, the number of hidden layers and the number of neurons in the layers were selected *a priori* rather arbitrarily and no attempts were made towards optimizing these selections. Since the aim is to prove the algorithm's local convergence performance and not to find an optimal structure for the neural networks, the structure chosen here is hoped to be complicated enough for the purpose of an extended observation. Clearly, for the neural networks, the input neurons represent their input variables, while the output neurons represent the output variables.

The training problems of the neural networks were solved using DE with control settings of $N_p = 20$, $F = 0.99$ and $C_r = 0.85$. The convergence histories for training the two neural networks averaged over 10 independent trial runs are shown in Figure 4. In cases for both practical neural network training problems studied, it is also observed that improve DE donors have a significantly higher convergence rate within the given limited evolving epoch in comparison with the original DE donor.

Because both the chosen neural network training problems were time consuming, the training processes could not be extended long enough to allow any of the compared algorithms to converge to the global optimum. However, in this article our attention was focused on how the

algorithms behave within the given maximum number of generations, that is, under tight time limitations that do not allow finding a globally optimal solution and therefore forces us to seek the best possible sub-optimal solution within the available computing time. The numerical simulation results for both studied neural networks, as shown in the Figure 4, demonstrates a clearly higher convergence rate with the new DE mutation donor schemes than with the original DE mutation donor scheme in training of the neural networks.

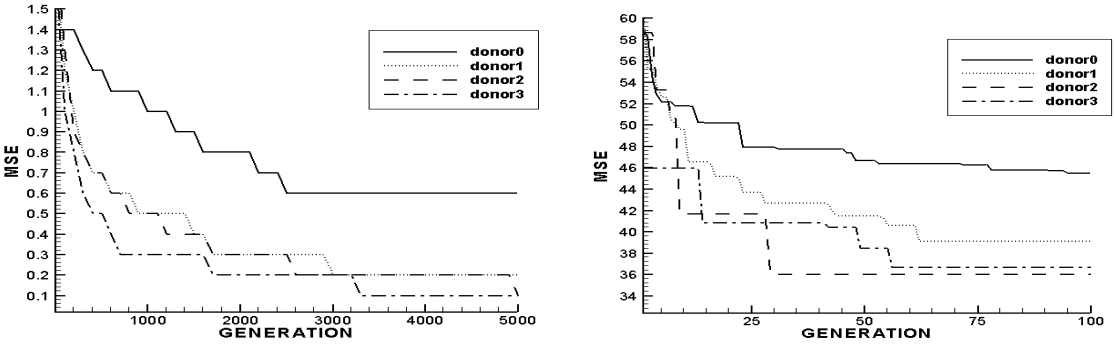

Figure 4: Convergence histories of DE with different mutation donor schemes in training the neural network: (a) for performance-prediction of a prototypic mixer; (b) for an aerodynamic five-hole probe calibration

6 CONCLUSIONS

In this article, three new mutation donor schemes are proposed for DE. The three schemes are empirically studied in comparison with the original mutation donor scheme by using a suite of six well-known test functions, and are also demonstrated by two practical application cases—training of neural networks to approximate aerodynamic data. The obtained numerical simulation results suggest that these modifications to the mutation operation can statistically improve DE's convergence performance.

Generally, when an EA is applied for solving a real-world problem, a "trade-off" has to be made between the algorithm's convergence rate and the solution accuracy. Often, the attempts towards a higher convergence rate will simultaneously increase the risk that the algorithm will terminate prematurely in a local optimum. In the light of the reported results, the proposed modifications to DE appear to satisfy both concerns. In the performed experiments all the three new mutation donor schemes were able to accelerate the DE convergence rate, without sacrificing the solution precision or robustness of the DE algorithm.

7 ACKNOWLEDGEMENTS

The authors would like to express their gratefulness to Mrs. Junhong Liu from the Department of Information Technology, LUT, Finland, for her assistance in carrying out the simulations. Prof. Y. Levy from Aerospace Engineering Faculty, Technion, Israel, provided his kind permission to use the related simulation data, which is also acknowledged with gratefulness.

REFERENCES

1. R. Storn and K. Price, *DE-a simple and efficient adaptive scheme for global optimization over continuous space*, Technical Report TR-95-012, ICSI, March (1995).
2. R. Storn and K. Price, "DE-a simple evolution strategy for fast optimization", *Dr. Dobb's Journal*, April 97, pp. 18-24 and 78 (1997).
3. R. Storn and K. Price, "DE-a simple and efficient heuristic for global optimization over continuous space", *Journal of Global Optimization*, 11(4), pp.341-359, December, (1997).
4. K. Price, "DE: a fast and simple numerical optimizer", *1996 Biennial Conference of the North American Fuzzy Information Processing Society*, NAFIPS, 19~22 June 1996, M. Smith, M. Lee, J. Keller, J. Yen, Eds., pp. 524-527, IEEE Press, New York, NY, (1996).
5. K. Price, "An Introduction to DE", David Corne, Marco Dorigo and Fred Glover, Eds., *New Ideas in Optimization*, McGraw-Hill, London (UK), pp. 78-108, (1999).
6. J. Lampinen, *A bibliography of differential evolution algorithm*, Technical Report, Lappeenranta University of Technology, Department of Information Technology, Laboratory of Information Processing, (2000).
7. R. Storn and K. Price, "Minimizing the real function of the ICEC'96 contest by DE", *IEEE International Conf. on Evolutionary Computation*, Nagoya, May, pp.842-844 (1996).
8. J. Lampinen and I. Zelinka, "Mechanical engineering design optimization by differential evolution", David Corne, Marco Dorigo and Fred Glover, Eds., *New Ideas in Optimization*, McGraw-Hill, London (UK), pp. 127-146 (1999).
9. J. Lampinen and I. Zelinka, "Mixed integer-discrete-continuous optimization by differential evolution, Part 1: the optimization method", Ošmera, Pavel, Ed., *Proceedings of MENDEL'99, 5th International Mendel Conference on Soft Computing*, Brno, Czech Republic, pp.71-76 (1999).
10. J. Lampinen and I. Zelinka, "Mixed integer-discrete-continuous optimization by differential evolution, Part 2: a practical example", Ošmera, Pavel, Ed., *Proceedings of MENDEL'99, 5th International Mendel Conference on Soft Computing*, Brno, Czech Republic, pp. 77-81 (1999).
11. J. Lampinen, "A Constraint Handling Approach for the Differential Evolution Algorithm", *The 2002 IEEE World Congress on Computational Intelligence WCCI 2002*, Honolulu, Hawaii, May 12-17, 2002. 6 pages. IEEE. ISBN 0-7803-7281-6 (CD-ROM).
12. K. De Jong, *An analysis of the behavior of a class of genetic adaptive systems*, Ph.D. Thesis, Department of Computer and Communication Sciences, University of Michigan, Ann Arbor, MI, (1975).
13. T. Back, *Evolutionary Algorithms in Theory and Practice*, Oxford University Press, Inc., Oxford, (1996).
14. H. Muhlenbein, M. Schomisch and J. Born, "The parallel genetic algorithm as function optimizer", *Parallel Computing*, 17, pp. 619~632, (1991).
15. H. Y. Fan and J. Lampinen, "A trigonometric mutation operation to differential evolution", *Journal of Global Optimization*, (2003). (in press)
16. Y. Levy, H. Y. Fan and V. Sherbaum, "A numerical investigation of mixing process of a novel combustor application", submitted to *ASME Journal of Heat Transfer*.