

A Directed Mutation Operator for Real Coded Genetic Algorithms

Imtiaz Korejo, Shengxiang Yang, and Changhe Li

Department of Computer Science, University of Leicester
University Road, Leicester LE1 7RH, UK
{iak5, s.yang, cl160}@mcs.le.ac.uk

Abstract. Developing directed mutation methods has been an interesting research topic to improve the performance of genetic algorithms (GAs) for function optimization. This paper introduces a directed mutation (DM) operator for GAs to explore promising areas in the search space. In this DM method, the statistics information regarding the fitness and distribution of individuals over intervals of each dimension is calculated according to the current population and is used to guide the mutation of an individual toward the neighboring interval that has the best statistics result in each dimension. Experiments are carried out to compare the proposed DM technique with an existing directed variation on a set of benchmark test problems. The experimental results show that the proposed DM operator achieves a better performance than the directed variation on most test problems.

1 Introduction

Genetic algorithms (GAs) are a class of probabilistic optimization techniques inspired by genetic inheritance and natural evolution. GAs have been used for solving many optimization problems due to the properties of self-learning, self-organization, and self-adaptation, as well as the properties of implicit parallelism [1, 5]. GAs are population based approaches, where new populations are generated by the iterative application of selection and variation of individuals in the population. For example, mutation is used to explore new solutions, crossover exchanges genetic information between two individuals, and selection selects relatively fit individuals for the next population. The fitness of an individual is evaluated by a fitness function, which defines the external environment of a GA. The performance of a GA depends on not only the above variation operators, but also some other factors, e.g., the population size and selection method, etc.

Mutation is a key operator to increase the diversity of the population and hence enables GAs to explore promising areas of the search space [10]. The step size and search direction are major factors that determine the performance of mutation operators. It may be beneficial to use different values during different stages of evolution in order to get a better performance of GAs. However, it is impossible to know the optimum mutation step size and search direction

for real-world problems. Hence, it is usually beneficial to use some statistics information to guide the mutation operator for GAs [4]. Strategy parameters are adjusted according to one of the three methods: deterministic adaptation adjusts the values of parameters according to predefined rules without using any learning information from GAs; adaptive adaptation alters the parameters using some learning information from the search space. The best example of adaptive adaptation is Rechenberg’s 1/5 success rule in evolutionary strategies; and self-adaptive adaptation embeds the parameters into the chromosomes of individuals and modifies the parameters by the GA itself.

Several researchers have tried to increase the performance of real-coded GAs by using directed mutation techniques [2, 6, 12]. In [3], the authors proposed a co-evolutionary technique where each component of a solution vector is added one extra bit to determine the direction of mutation. The direction bit is adapted by using the feedback information from the current population. A directed mutation based on momentum was proposed in [12], where each component of an individual is attached a standard gaussian mutation and the current momentum to mutate that component.

In this paper, a new directed mutation technique is proposed for GAs to explore promising areas in the search space. This approach first calculates the statistical information from the current population regarding the average fitness and the number of individuals distributed within each interval of each dimension of the search space. Then, the statistical information is used to guide the mutation of an individual toward the neighboring interval that has a better statistics result in each dimension. In order to investigate the performance of the proposed directed mutation, an experimental study is carried out to compare the performance of the GA with the proposed directed mutation and the GA with the directed mutation in [2].

2 Related work

A directed variation (DV) technique was proposed by Zhou and Li [14]. This algorithm does not introduce a scheme for completely generating an average step size but adjusts some individuals by using the feedback information from the current population. Suppose the population is a set of N individuals $X = \{\vec{x}_1, \vec{x}_2, \dots, \vec{x}_N\}$ and each individual is a K -dimensional vector, denoted by $\vec{x}_i = [x_{i1}, x_{i2}, \dots, x_{iK}]$. Denote the minimal d -th component of the individuals at generation t by x_d^L and the maximum by x_d^U , that is, the range of the d -th dimension at time t is $R_d(t) = [x_d^L, x_d^U]$. This range can be equally divided into L intervals. The fitness of a nonempty interval, say, the j -th interval of the d -th dimension, is defined by:

$$F_{dj} = \sum_{i=1}^N I(x_{id} \in B_{dj}) f_{Norm}(\vec{x}_i) \quad (1)$$

$$I(x_{id} \in B_{dj}) = \begin{cases} 1, & \text{if } x_{id} \in B_{dj} \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

where B_{dj} denotes the range (lower and upper bounds) of the j -th interval of the d -th dimension, N presents the population size, $I(\cdot)$ is the indicator function, and the fitness of each solution vector \vec{x}_i is normalized as follows:

$$f_{Norm}(\vec{x}_i) = \frac{f(\vec{x}_i) - f_{min}}{f_{max} - f_{min}} \quad (3)$$

where f_{max} and f_{min} represent the maximum and minimum fitness of the whole population respectively.

With DV, in each generation some individuals are selected for directed variation in each component. DV is applied on a component, say, the d -th component, only when the range of the d -th dimension of all solutions in the current generation decreases in comparison with that of the previous generation, i.e., the population converges regarding that dimension. DV works as follows. First, the fitness of interval, i.e., F_{dj} , is calculated according to Eq. (1). Then, DV is applied for an individual component by component, where each component of the individual may be shifted from its current interval to a neighboring interval that has a higher fitness with a certain probability, as described below.

In DV, for each component $x_{id} \in B_{dj}$ of an individual \vec{x}_i , whether it is mutated depends on the value F_{dj} and the fitness of its neighboring intervals, i.e., $F_{d,j-1}$ and $F_{d,j+1}$. If F_{dj} is bigger than both $F_{d,j-1}$ and $F_{d,j+1}$, then DV is not applied to the d -th component of any selected individuals with $(x_{id}) \in B_{dj}$. If F_{dj} is in the middle, without loss of generality, suppose $F_{d,j-1} > F_{dj} > F_{d,j+1}$, the probability of directed variation, P_{dj}^{DV} , can be calculated as follows:

$$P_{dj}^{DV} = 1 - \frac{F_{dj}}{F_{d,j-1}} \quad (4)$$

With this probability, x_{id} is replaced with a number, randomly generated between x_{id} and the center of $B_{d,j-1}$. If F_{dj} is smaller than both $F_{d,j-1}$ and $F_{d,j+1}$, then either $B_{d,j-1}$ or $B_{d,j+1}$ is randomly selected with an equal probability and x_{id} moves towards the selected interval, i.e., replaced with a number randomly generated between x_{id} and the center of the selected interval.

3 Directed mutation for genetic algorithms

The main motivation behind directed mutation (DM) is to explore promising areas in the search space by using the feedback information from the current population, e.g., the fitness and some other factors. It is a modified version of the standard mutation. This paper introduces a DM technique which aims to explore promising areas of the search space with fixed boundaries according to the fitness of intervals and the percentage of individuals in each interval of each dimension. The framework of the GA with the proposed DM operator is given in Algorithm 1. The proposed GA differs from the standard GA in that in each generation, a set of individuals are selected to undergo the DM operation iteratively. As shown in Algorithm 2, the DM operator is applied for each component of a selected

Algorithm 1 GA with Directed Mutation (DM)

- 1: Randomly generate an initial population pop
- 2: Evaluate the fitness of each individual of pop
- 3: $t := 0$.
- 4: **while** $t < max_gen$ **do**
- 5: **for** each individual i in pop **do**
- 6: Select individual j by the roulette wheel method
- 7: Crossover individual i with individual j using the arithmetic crossover method
- 8: Mutate individual i by using the Gaussian mutation with mean zero and pre-selected or adaptive standard deviation
- 9: **end for**
- 10: Apply DM on a set of individuals randomly selected from the population
- 11: Generate next generation by the roulette wheel selection scheme
- 12: $t := t + 1$
- 13: **end while**

Algorithm 2 Directed Mutation

- 1: **for** each dimension $d \in \{1, 2, \dots, K\}$ **do**
- 2: **if** $R_d(t) < R_d(t - 1)$ **then**
- 3: **for** each interval j **do**
- 4: Calculate F_{dj} according to Eq. (1)
- 5: Calculate the number of individuals P_{dj} according to Eq. (5)
- 6: **end for**
- 7: **for** each interval j **do**
- 8: Calculate FP_{dj} according to Eq. (6)
- 9: **end for**
- 10: **for** each interval j **do**
- 11: Calculate P_{dj}^{DM} according to Eq. (7)
- 12: **end for**
- 13: Shift the genes x_{id} of selected individuals to their neighboring interval with a higher fitness with the associated probability
- 14: **end if**
- 15: **end for**

solution in a similar way as the DV operator described above. The difference lies in the calculation of the probability of moving a component of a solution from one interval to its neighboring interval, which is described in detail below.

Similar to the DV operator, the range of the d -th dimension of individuals at generation t , i.e., $R_d(t) = [x_d^L, x_d^U]$, is also equally divided into L intervals. The fitness F_{dj} of each non-empty interval is calculated by Eq. (1), Eq. (2), and Eq. (3). In addition to the fitness of each interval, the percentage of individuals in each interval is also calculated in the DM operator as follows:

$$P_{dj} = \frac{1}{N} \sum_{i=1}^N I(x_{id} \in B_{dj}) \quad (5)$$

where P_{dj} represents the percentage of individuals with the d -th component in the j -th interval in the current population, and B_{dj} and $I(\cdot)$ are the same as defined before in Eq. (1) and Eq. (2). From F_{dj} and P_{dj} , we calculate a value that is associated with the j -th interval of the d -th dimension, assuming $F_{d,j-1} > F_{dj} > F_{d,j+1}$, as follows:

$$FP_{dj} = \frac{F_{dj}}{F_{d,j-1}} + \frac{P_{dj}}{P_{d,j-1}} \quad (6)$$

With above definitions, the DM operator works to mutate a component x_{id} of a selected individual \vec{x}_i as follows. DM will not be used to x_{id} if F_{dj} is bigger than the fitness of both neighboring intervals, i.e., $F_{d,j-1}$ and $F_{d,j+1}$. If FP_{dj} is in the middle in comparison with the fitness of its two neighbor intervals $j-1$ and $j+1$, without loss of generality, suppose $FP_{d,j-1} > FP_{dj} > FP_{d,j+1}$. Then, move the individual \vec{x}_i towards the interval $j-1$ with a certain probability, which is calculated as follows.

$$P_{dj}^{DM} = \frac{FP_{dj}}{\sum_{j=1}^L FP_{dj}} \quad (7)$$

where the DM probabilities P_{dj}^{DM} are normalized over all intervals. In this case, the solution \vec{x}_i is moved toward the interval $j-1$ by replacing x_{id} with a number randomly generated between x_{id} and the center of $B_{d,j-1}$ as follows:

$$x_{id} = rand(x_{id}, B_{d,j-1}) \quad (8)$$

Otherwise, if $FP_{dj} < FP_{d,j-1}$ and $FP_{dj} < FP_{d,j+1}$, then either $B_{d,j-1}$ or $B_{d,j+1}$ is selected with an equal probability and the solution \vec{x}_i moves towards the selected interval with the probability P_{dj}^{DM} .

The main motivation behind the proposed DM operator lies in that moving individuals based on Eq. (6) and Eq. (7) can help the GA explore the promising intervals in the search space. Since individuals in DV only move toward the interval with the highest fitness, it may easily cause the premature convergence problem.

In the proposed DM operator, individual shifting is not only based on the feedback information of the average fitness of intervals, but also on the population distribution. By taking into account the information of population distribution, DM efficiently avoids the premature convergence problem. The key idea of the DM operator is illustrated in Fig. 1.

From Fig. 1, we consider the j -th interval, if we only consider DV, the two individuals of interval j will move toward the $(j-1)$ -th interval due to the higher fitness of the $(j-1)$ -th interval. However, the right direction should be the $(j+1)$ -th interval since the $(j+1)$ -th interval is more promising than the $(j-1)$ -th interval. Hence, DM is an enhanced version of DV.

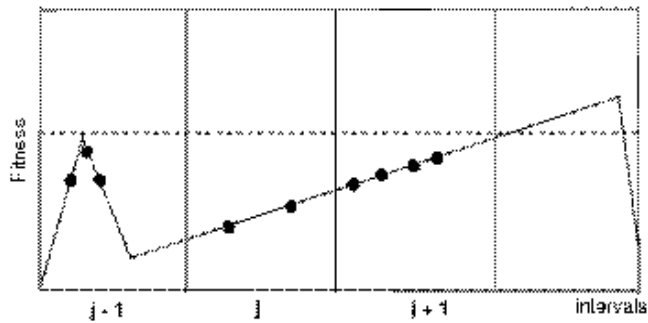


Fig. 1. Fitness landscape of the d -th dimension

4 Experimental study

4.1 Experimental setting

In order to test the performance of GA with DM, three unimodal functions and eleven multimodal functions, which are widely used as the test functions in the literature [11, 13], were selected as the test bed in this paper. The number of dimensions n is set to 10 for all test functions. The details of these test functions are given in Table 1. Function f_9 is a composition function proposed by Suganthan et al. [11], which is composed of ten benchmark functions: the rotated version and shifted version of $f_1, f_2, f_3, f_4,$ and f_5 , as also listed in Table 1, respectively. Functions f_{10} to f_{14} are rotated functions, where the rotation matrix M for each function is obtained using the method in [8].

The idea of DV was taken from [14], which is implemented in the peer GA. The adaptive standard deviation [9] is used in DM. The population size (100) and total number of generations (500) are the same for both DM and DV on all problems, the total number of intervals L was set to 3, 6, 9, and 12, respectively. The mutation probability P_m for the Gaussian mutation is the same for DM and DV, which was set to different values for different test problems, as listed in Table 1. Both the GA with DM and the GA with DV were run 30 times independently on each test problem.

4.2 Experimental results and analysis

The average results of 30 independent runs of the GA with directed mutation and the GA with directed variation on the test problems are shown in Table 2. From Table 2, it can be seen that the number of intervals used for each dimension is a key parameter. The performance of the GA with DM becomes significantly better on some problems than that of the GA with DV as the number of intervals increases. The performance of both operators is different on different problems.

Table 1. Test functions of $n = 10$ dimensions, where D ($D \in R_n$) and f_{min} denote the domain and the minimum value of a function respectively

Test Function	P_m	D	f_{min}
$f_1(x) = \sum_{i=1}^n x_i^2$	0.1	[-100, 100]	0
$f_2(x) = \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i) + 10)$	0.01	[-5.12, 5.12]	0
$f_3(x) = \sum_{i=1}^n \sum_{k=0}^{k_{max}} [a^k \cos(2\pi b^k (x_i + 0.5))] - n \sum_{k=0}^{k_{max}} [a^k \cos(\pi b^k)]$, $a = 0.5, b = 3, k_{max} = 20$	0.01	[-0.5, 0.5]	0
$f_4(x) = \frac{1}{4000} \sum_{i=1}^n (x_i - 100)^2 - \prod_{i=1}^n \cos(\frac{x_i - 100}{\sqrt{i}}) + 1$	0.01	[-600, 600]	0
$f_5(x) = -20 \exp(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}) - \exp(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)) + 20 + e$	0.01	[-32, 32]	0
$f_6(x) = \sum_{i=1}^n 100(x_{i+1}^2 - x_i)^2 + (x_i - 1)^2$	0.05	[-30, 30]	0
$f_7(x) = \sum_{i=1}^n -x_i \sin(\sqrt{ x_i })$	0.01	[-500, 500]	-4189.829
$f_8(x) = \sum_{i=1}^n (\sum_{j=1}^i x_j)^2$	0.01	[-100, 100]	0
$f_9(x) =$ Composition function (CF 5) in [11]	0.05	[-5, 5]	0
$f_{10}(x) = \sum_{i=1}^n 100(y_{i+1}^2 - y_i)^2 + (y_i - 1)^2, \mathbf{y} = \mathbf{M} * \mathbf{x}$	0.05	[-100, 100]	0
$f_{11}(x) = \frac{1}{4000} \sum_{i=1}^n (y_i - 100)^2 - \prod_{i=1}^n \cos(\frac{y_i - 100}{\sqrt{i}}) + 1,$ $\mathbf{y} = \mathbf{M} * \mathbf{x}$	0.01	[-600, 600]	0
$f_{12}(x) = -20 \exp(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n y_i^2}) - \exp(\frac{1}{n} \sum_{i=1}^n \cos(2\pi y_i)) + 20 + e, \mathbf{y} = \mathbf{M} * \mathbf{x}$	0.01	[-32, 32]	0
$f_{13}(x) = \sum_{i=1}^n (y_i^2 - 10 \cos(2\pi y_i) + 10), \mathbf{y} = \mathbf{M} * \mathbf{x}$	0.01	[-5, 5]	0
$f_{14}(x) = \sum_{i=1}^n \sum_{k=0}^{k_{max}} [a^k \cos(2\pi b^k (y_i + 0.5))] - n \sum_{k=0}^{k_{max}} [a^k \cos(\pi b^k)]$, $a = 0.5, b = 3, k_{max} = 20, \mathbf{y} = \mathbf{M} * \mathbf{x}$	0.01	[-0.5, 0.5]	0

When the number of intervals is set to 3, the results of DM are better than DV on half of the test problems. DM is trapped into local optima due to the large range of intervals. It is interesting that DM achieves the best result on f_9 , f_{10} , and f_{12} over all different number of intervals.

When we increase the number of intervals to 6, the performance of DM is better than DV on most benchmark problems. Although DV presents better results than DM on f_1 , f_2 , f_9 , and f_{14} , DM obtains close results to DV on these functions.

Similar observations can be made as the number of intervals increases to 9. The results obtained by DM are better than that of DV. Compared with the results of DM with the number of intervals of 6, the performance of DM deteriorates on some multimodal problems. However, the results of DM with the number of intervals of 9 are better than the results with the number of intervals of 6 on most unimodal problems.

Table 2. Comparison results between DV and DM with the number of intervals for each dimension set to different values for different problems

function		f_1	f_2	f_3	f_4	f_5	f_6	f_7
$L = 3$	DV	8.28e-06	0.0151	0.2590	0.0985	0.0088	49.41	-2667
	DM	7.33e-06	0.0122	0.2304	0.1116	0.0082	33.87	-2062
$L = 6$	DV	7.55e-06	0.0093	0.2030	0.0820	0.0114	95.25	-2692
	DM	8.99e-06	0.0172	0.1837	0.0385	0.0090	18.84	-1797
$L = 9$	DV	6.19e-06	0.0087	0.2181	0.0731	0.0076	68.78	-2574
	DM	9.03e-06	0.0067	0.2558	0.1143	0.0089	22.97	-1857
$L = 12$	DV	7.21e-06	0.0191	0.2271	0.0972	0.0107	16.63	-2578
	DM	7.86e-06	0.0153	0.2196	0.0341	0.0080	5.65	-1905
t -test	DV-DM	~	~	~	~	~	+	-
function		f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}
$L = 3$	DV	2.7677	102	61	0.0929	0.0476	2.0971	0.4200
	DM	3.5671	100	21	0.0891	0.0093	2.8655	0.4794
$L = 6$	DV	2.6971	64	82	0.1147	0.0470	2.5969	0.4283
	DM	1.9957	94	30	0.0726	0.0089	2.1034	0.3483
$L = 9$	DV	3.8969	94	30	0.0726	0.0974	2.1034	0.4099
	DM	3.8160	24	24	0.0998	0.0077	2.7950	0.4833
$L = 12$	DV	2.0960	230	50	0.0826	0.0472	3.0315	0.4175
	DM	2.7178	63	53	0.0444	0.0086	1.9218	0.4180
t -test	DV-DM	~	+	+	~	+	~	~

When $L = 12$, the results of DM are better than the results of DV on most test functions. Similar results can be viewed as the number of interval 6 but the performance of DM increases compared with the number of interval of 9 on some multimodal problems.

Table 2 also shows the statistical analysis of comparing DM with DV when $L = 6$ by using the two-tailed t -test with a 58 degree of freedom at a 0.05 level of significance, where the t -test result is presented as “+”, “-”, or “~” if the performance of the GA with DM is significantly better than, significantly worse than, or statistically equivalent to the GA with DV, respectively. The DM operator is significantly better on four problems, significantly worse on two problems, and statistically similar on the rest of the problems.

From Table 2, three conclusions can be made. First, the overall performance of DM is better than DV on half test functions at least. Especially, on f_9 , f_{10} and f_{12} , the performance of DM is better over all different settings of the number of intervals. Second, the interval quantity is a crucial factor to the performance of both DM and DV on different benchmark functions. According to variable number of intervals, the result of DM and DV varies on different test problems. Third, a larger number of intervals is needed on multimodal problems than unimodal problems. The smaller number of intervals causes the larger number of local optima within an interval, since multimodal problems have many local optima.

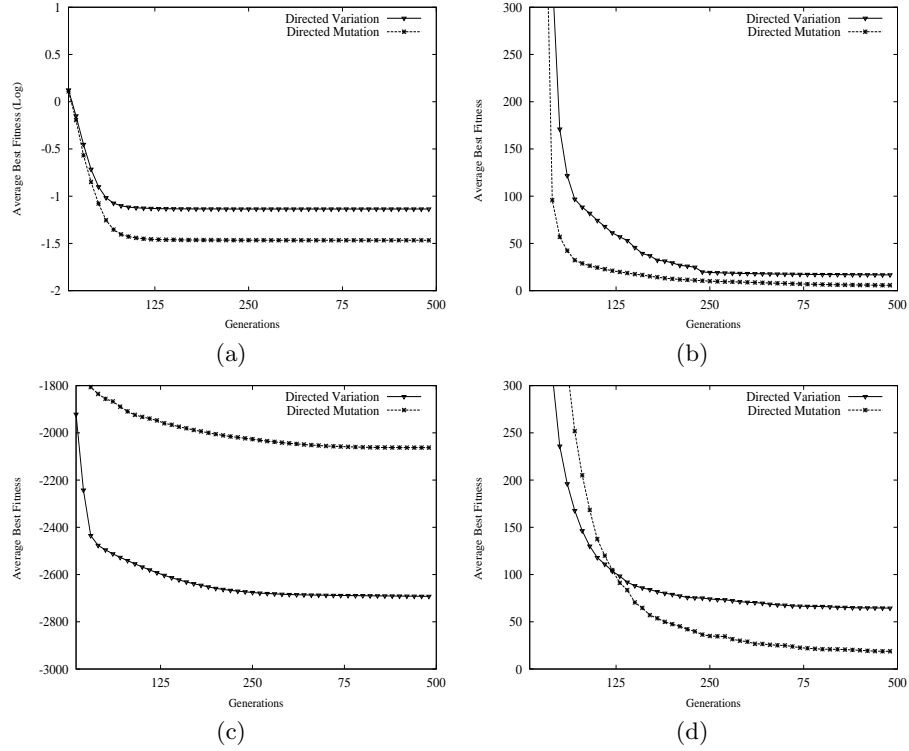


Fig. 2. Evolutionary progress of directed mutation and directed variation operators on (a) f_4 with $L = 12$, (b) f_6 with $L = 12$, (c) f_7 with $L = 3$, and (d) f_9 with $L = 6$.

Fig. 2 presents the evolutionary process for DM and DV operators on f_4 , f_6 , f_7 , and f_9 respectively, where the result on f_4 is presented in a log scale. From Fig. 2, it can be seen that the convergence speed of DM is faster than that of DV except on f_9 . This result validates our idea that the performance of DV can be enhanced by taking into account the population distribution in the search space.

5 Conclusions

In this paper, a directed mutation operator is proposed for genetic algorithms to explore promising solutions in the search space. In the proposed directed mutation, individual shifting is not only based on the feedback information of the fitness of each interval, but also on the population distribution. By taking into account the information of the population distribution, directed mutation greatly improves the performance of directed variation.

In order to justify the proposed directed mutation, a set of benchmark functions was used as the test base to compare the performance of the directed mutation operator with the directed variation operator from the literature [14]. The experimental results show that the efficiency of DV is improved by the proposed enhancement on more than half of the test functions.

Although DM achieves better results on the test problems, there is a limitation, i.e., different problems need different optimum values of the number of intervals to achieve the best result. So, how to adaptively adjust the number of intervals is our major work in the future.

References

1. T. Back. *Evolutionary Algorithms in Theory and Practice*, Oxford University Press, 1996.
2. S. Berlik. A directed mutation framework for evolutionary algorithms, *Proc. of the Int. Conf. on Soft Computing, MENDEL*, 2004.
3. A. Berry and P. Vamplew. PoD Can Mutate: A simple dynamic directed mutation approach for genetic algorithms, *Proc. of AISAT 2004: Int. Conf. on Artificial Intelligence in Science and Technology*, pp. xxx-yyy, 2004.
4. A. E. Eiben, Z. Michalewics, M. Schoenauer, and J. E. Smith. Parameter control in evolutionary algorithms. In xxxxx (eds.), *Parameter Setting in Evolutionary Algorithms*, Chapter y, pp. 19-46, 2007.
5. D. E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*, New York: Addison Wesley, 1989.
6. A. R. Hedar and M. Fukushima. Directed evolutionary programming: Towards an improved performance of evolutionary programming. *Proc. of the 2006 IEEE Congress on Evol. Comput.*, pp. 1521-1528, 2006.
7. F. Herrera and M. Lozano. Two-loop real-coded genetic algorithms with adaptive control of mutation step size. *Applied Intelligence*, 13: 187-204, 2002.
8. R. Salomon. Reevaluating genetic algorithm performance under coordinate rotation of benchmark functions: A survey of some theoretical and practical aspects of genetic algorithms. *BioSystems*, 39(3): 263-278, 1996.
9. Schwefel. H-P. *Evolution and Optimum Seeking*, New York: Wiley, 1995.
10. W. M. Spears. Crossover or mutation? In L. D. Whitley (ed.), *Foundations of Genetic Algorithms 2*, San Mateo: Morgan Kaufmann publishers, pp. 81-89, 1993.
11. P. N. Suganthan, N. Hansen, J. J. Liang, K. Deb, Y.-P. Chen, A. Auger, and S. Tiwari. Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization. *Technical Report*, Nanyang Technological University, Singapore, 2005.
12. L. Temby, P. Vamplew, and A. Berry. Accelerating real valued genetic algorithms using mutation-with-momentum. *AI 2005: Advances in Artificial Intelligence*, LNCS 3809, pp. 1108-1111, 2005.
13. X. Yao, Y. Liu and G. Lin. Evolutionary programming made faster. *IEEE Trans. on Evol. Comput.*, 3(2): 82-102, 1999.
14. Q. Zhou and Y. Li, Directed variation in evolutionary strategies. *IEEE Trans. on Evol. Comput.*, 7(4): 356-366, 2003.