Adaptive Mutation with Fitness and Allele Distribution Correlation for Genetic Algorithms

Shengxiang Yang
Department of Computer Science
University of Leicester
University Road, Leicester LE1 7RH, UK
s.yang@mcs.le.ac.uk

Şima Uyar
Istanbul Technical University
Computer Engineering Department
Maslak 34469 Istanbul, Turkey
etaner@cs.itu.edu.tr

ABSTRACT

In this paper, a new gene based adaptive mutation scheme is proposed for genetic algorithms (GAs), where the information on gene based fitness statistics and on gene based allele distribution statistics are correlated to explicitly adapt the mutation probability for each gene locus over time. A convergence control mechanism is combined with the proposed mutation scheme to maintain sufficient diversity in the population. Experiments are carried out to compare the proposed mutation scheme to traditional mutation and two advanced adaptive mutation schemes on a set of optimization problems. The experimental results show that the proposed mutation scheme efficiently improves GA's performance.

Categories and Subject Descriptors

I.2.8 [**Computing Methodologies**]: Artificial Intelligence—*Problem Solving, Control Methods, and Search*

General Terms

Algorithms, Experimentation, Performance

Keywords

Genetic algorithms, gene based adaptive mutation, fitness and allele distribution correlation

1. INTRODUCTION

Genetic algorithms (GAs) are one class of probabilistic optimization approaches inspired by the principles of natural evolution. The performance of GAs depends on many factors, such as selection and reproduction schemes. This makes it difficult, if not impossible, to choose operators and relevant parameters for optimal performance. As one key ingredient of GAs, mutation works by changing individuals bitwise with a small probability $p_m \in [0, 1]$ and is used to ensure that all possible alleles can enter the population and hence maintain the population diversity. Many papers, both practical [8] and theoretical [10], have been devoted to study the nature of mutation, of which many have been investigated on finding globally "optimal" p_m for GAs [1, 7].

However, there has been further studies [2, 11] showing that the optimal p_m will not only depend on the problem being solved but also vary with the progress of searching. Hence, developing adaptive mutation rate strategies may not only overcome the difficulties of finding "optimal" settings but adapt the mutation rate well according to the evolutionary process. Generally speaking, adaptation in mutation happens at two levels. At the top level, the ratio between mutation and crossover is adapted during the run of a GA [6, 12]; At the bottom level, the mutation probability is adapted during the run of a GA, uniformly or non-uniformly over the loci [1, 3, 5]. In [13, 14], two adaptive mutation schemes were proposed, which adjust the mutation rate for each gene locus over time by the information of gene-based allele distribution and gene-based fitness statistics respectively.

In this paper, a new gene based adaptive mutation scheme is proposed for GAs, which combines the ideas behind the two adaptive mutation schemes in [13, 14]. In the proposed mutation scheme, the statistics information on gene based fitness and gene based allele distribution in each gene locus are correlated to explicitly adapt the mutation probability for that locus. And hence the mutation probability for each gene locus is adapted more accurately toward the problem in hand and the evolutionary searching progress. A convergence control mechanism, which selects a portion of individuals from the population to be complemented when the population has converged to certain degree, is combined with the proposed mutation scheme to maintain sufficient diversity in the population.

2. RELATED WORK

2.1 Statistics-based Adaptive Non-Uniform Mutation

In paper [14], a *Statistics-based Adaptive Non-Uniform Mutation* (SANUM) was proposed. For the convenience of description, we first describe the concepts of intrinsic attribute and extrinsic tendency of allele valuing for a gene locus. In a binary-encoded solution for a given problem, a gene locus is called *1-intrinsic* if its allele is 1, *0-intrinsic* if its allele is 0, or *neutral* if its allele can be either 0 or 1. Whether a locus is 1-intrinsic, 0-intrinsic, or neutral depends on the problem being solved and encoding scheme. During the run of a GA, a gene locus is called *1-inclined* if the frequency of 1's in its alleles over the population tends

SAC'06, April 23–27, 2006, Dijon, France.

to increase (to the limit of 1.0) with time, *0-inclined* if the frequency of 1s tends to decrease (to the limit of 0.0), or *non-inclined* if there is no tendency of increasing or decreasing. Usually with the progress of the GA, those gene loci that are 1-intrinsic (or 0-intrinsic) will appear to be 1-inclined (or 0-inclined), i.e., the frequency of 1's in the alleles of these loci will eventually converge to 1 (or 0). SANUM uses this convergence information as feedback to adapt the mutation probability for each locus.

We use the frequency of 1's in the alleles in a locus of the population to calculate corresponding mutation probability of that locus. The frequency of 1's in the alleles of a locus can be taken as the degree of convergence to "1" for that locus. Let $f_1(i,t)$ $(i = 1, \ldots, L)$ denote the frequency of 1's in the alleles in locus i over the population at generation t and $p_m(i,t)$ $(i = 1, \ldots, L)$ denote the mutation probability of locus i at generation t. In [14], a linear formula is used to calculate $p_m(i,t)$ from $f_1(i,t)$. In this paper, a Gaussian function is used to calculate $p_m(i,t)$ as follows.

$$p_m(i,t) = \alpha * exp(-(f_1(i,t) - 0.5)^2/\beta), \quad (1)$$

where α controls the maximum allowable mutation probability for a locus and β is the parameter that controls the falling speed of $p_m(i,t)$ when $f_1(i,t)$ diverts away from 0.5. The smaller the value of β, the faster the falling speed of $p_m(i,t)$, and the stronger the protection to converged genes or building blocks found so far. In SANUM with the Gaussian function, a lower bound P_{min} can be applied to each mutation probability by limiting it within $[P_{min}, \alpha]$, i.e., $p_m(i,t) \in [P_{min}, \alpha]$ for each locus i.

2.2 Gene Based Adaptive Mutation

In paper [13], a *Gene Based Adaptive Mutation* (GBAM) was proposed, which uses an asymmetric adaptive mutation mechanism on each locus of the chromosome. In GBAM, there are two different mutation rates defined for each locus: p_m^1 for those genes that have alleles of "1" and p_m^0 for those genes that have alleles of "0". In the mutation operations, the appropriate mutation rate is applied based on the gene allele value. Initially, all the mutation probabilities are set to an initial value, e.g., 0.02. Then for each generation, the mutation probabilities p_m^0 and p_m^1 for each locus are updated based on the feedback statistics taken from the current population with respect to the relative success or failures of those individuals having a "1" or "0" at that locus. For example, for a maximization problem, the update rule for the two mutation probabilities for locus i at generation $t+1$ is given in Eqs. 2 and 3 as below.

$$p_m^0(i,t+1) = \begin{cases} p_m^0(i,t) + \gamma, & \text{if } G_{avg}^1(i,t) > P_{avg}(t) \\ p_m^0(i,t) - \gamma, & \text{otherwise} \end{cases} \quad (2)$$

$$p_m^1(i,t+1) = \begin{cases} p_m^1(i,t) - \gamma, & \text{if } G_{avg}^1(i,t) > P_{avg}(t) \\ p_m^1(i,t) + \gamma, & \text{otherwise,} \end{cases} \quad (3)$$

where γ is the mutation rate update value (a constant preset parameter), $G_{avg}^1(i,t)$ is the average fitness of individuals with an allele "1" for locus i at generation t, and $P_{avg}(t)$ is the average fitness of the population at generation t. This update rule is applied for each locus separately.

The main motivation behind GBAM is to adapt the mutation rate for each locus based on the relative success or failure of individuals in the population with respect to that locus. If the individuals with a specific allele value for a locus is successful, i.e., their average fitness is greater than the average fitness of the whole population, the mutation rate regarding that specific allele value will be decreased in order to protect that allele value from being mutated (and hence destroyed). For example, for the maximization problem, if $G_{avg}^1(i,t) > P_{avg}(t)$, the allele value "1" for locus i is assumed to generate more successful results. Therefore, $p_m^1(i,t+1)$ is decreased and $p_m^0(i,t+1)$ is increased by γ.

In GBAM, the mutation rates are bounded, i.e., $p_m^0(i,t) \in [P_{min}, P_{max}]$ and $p_m^1(i,t) \in [P_{min}, P_{max}]$ for each locus i. If an update causes a mutation rate to exceed the limit it is set to the corresponding boundary value.

3. GENE BASED ADAPTIVE MUTATION WITH FITNESS AND ALLELE DISTRIBUTION CORRELATION

Both SANUM and GBAM are based on the assumption that during the run of GAs 1-inclined genes or genes with better performed allele with value "1" will be 1-intrinsic and hence need protection by decreasing corresponding mutation rates. Usually, this assumption holds for many problems and the adaptive schemes improve GA's performance. However, for some problems where deception and epistasis pervade this assumption may not hold and hence may mislead the GA into local optima quickly. In order to overcome this difficulty, a new gene based adaptive mutation that make use of the fitness and allele distribution correlation information, denoted *GBAM_FAD*, is proposed in this paper.

As in GBAM, in GBAM_FAD we also define two mutation probabilities p_m^0 and p_m^1 for each gene locus with the same meanings. In the mutation operations, the appropriate mutation rate is applied based on the gene allele value. And similarly all the mutation probabilities are initialized to a pre-set value. For each generation, the mutation probabilities for each locus are updated based on the correlated statistics taken from the current population with respect to the relative success of those individuals having a specific allele and the allele distribution (measured by the allele frequency) at that locus. For example, for a maximization problem, the update rule for the mutation rates in locus i at generation $t+1$ is given in Eqs. 4 and 5 as below:

$$p_m^0(i,t+1) = p_m^0(i,t) + \gamma * sgn((G_{avg}^1(i,t) - P_{avg}(t))(f_1(i,t) - 0.5)) \quad (4)$$

$$p_m^1(i,t+1) = p_m^1(i,t) - \gamma * sgn((G_{avg}^1(i,t) - P_{avg}(t))(f_1(i,t) - 0.5)) \quad (5)$$

where $f_1(i,t)$ is the frequency of 1's in the alleles in locus i over the population at generation t, γ, $G_{avg}^1(i,t)$, and $P_{avg}(t)$ have the same definitions as in Eqs. 2 and 3, and the sign function $sgn(x)$ returns 1, 0, or -1 for $x > 0$, $x = 0$ and $x < 0$ respectively.

The key idea behind GBAM_FAD is to correlate the relative success/failure information of individuals in the population in each locus with the allele distribution statistics. Here, we call gene locus i *1-successful* at time t if $G_{avg}^1(i,t) > P_{avg}(t)$ or *0-successful* if $G_{avg}^1(i,t) < P_{avg}(t)$). If a gene locus i is detected as 1-successful and appears to be 1-inclined, it means gene i is very likely to be 1-intrinsic and hence we need to protect the allele "1" for gene i by decreasing $p_m^1(i,t)$ and increasing $p_m^0(i,t)$. Symmetrically, the logic holds for 0-successful and 0-inclined genes. In both cases, the $sgn()$ function in Eqs. 4 and 5 will return a value of 1. However,

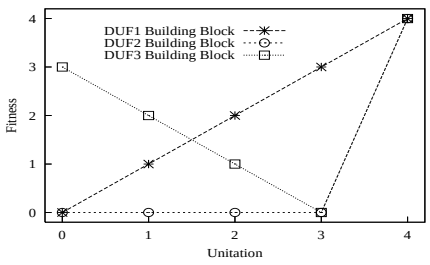

Figure 1: The building blocks for the three DUFs.

if a gene locus i is detected as 1-successful but appears 0-inclined, or vice versa, it means that conflict happens, e.g., due to deception or epistasis. In this case, $p_m^1(i,t)$ is increased while $p_m^0(i,t)$ is decreased in order to help the GA escape the potential local optima. Similarly, in GBAM_FAD the mutation probabilities can be limited within a range, i.e., $p_m^0(i,t) \in [P_{min}, P_{max}]$ and $p_m^1(i,t) \in [P_{min}, P_{max}]$.

4. TEST PROBLEMS

4.1 Decomposable Unitation-Based Functions

Decomposable unitation-based functions (DUFs), such as trap and deceptive functions, have been widely studied in GA's community in order to understand what constructs difficult problems for GAs [4]. A unitation function of a binary string returns the number of ones in the string. In this paper, three DUFs, denoted *DUF1*, *DUF2*, and *DUF3*, are constructed as the test functions. All the four DUFs consist of 25 copies of 4-bit building blocks (BBs). Each BB of the DUFs is a unitation-based function and contributes a maximum value of 4 to the total fitness, as shown in Figure 1. The fitness of a string is the sum of contributions from all BBs, giving an optimal fitness of 100 for all DUFs.

DUF1 is in fact an $OneMax$ function, aiming to maximize the number of ones in a string. For DUF2, in the search space of the 4-bit BB, the unique optimal solution is surrounded by all the other 15 solutions with zero fitness (needle-in-a-haystack), which makes it much harder for GAs than DUF1. And DUF3 is a fully deceptive function. Fully deceptive functions are usually hard problems for GAs [4].

4.2 The Four-Peaks Problem

The 100-bit 4-Peaks problem is defined as follows.

$$REWARD = \begin{cases} 100 + T, \text{ if } o(x) > T \text{ and } z(x) > T \\ 0, \text{ otherwise} \end{cases} \quad (6)$$

$$f(x) = max\{o(x), z(x)\} + REWARD, \quad (7)$$

where $o(x)$ is the number of contiguous ones at the head of the string (i.e., starting at position 1), $z(x)$ is the number of contiguous zeros at the end of the string (i.e., ending at position 100), and T is a threshold parameter. The 4-Peaks problem has four peaks: two global optima with fitness value being 199 and two local optima with fitness 100. With the increasing of the value of T, the basins of the attraction surrounding the inferior local optima increase in size exponentially while the basins around the global optima decrease at the same speed. Hence, increasing T makes it harder for a GA to escape the local optima. In this paper we set the value of T to 11 and 27 respectively.

5. EXPERIMENTAL STUDY

5.1 Experimental Settings

In order to test the performance of proposed GBAM_FAD, in this experimental study it is compared with traditional bit flip mutation with a "standard" probability $p_m = 1/L = 0.01$ [7], SANUM and GBAM. For SANUM, the parameters are set as: $(\alpha, \beta) = (0.05, 0.04)$ and $P_{min} = 0.0001$ (i.e., $p_m(i,t) \in [0.0001, 0.05]$ for each locus i). For GBAM and GBAM_FAD, the parameters are set as: $\gamma = 0.001$, $[P_{min}, P_{max}] = [0.0001, 0.2]$, and initially $p_m^1(i,0) = p_m^0(i,0) = 1/L = 0.01$ for each locus i. For all the GAs, other generators are set as follows: the tournament selection with tournament size of 2, elitism of size 1, 2-point crossover with a typical probability 1.0, and the population size $N = 250$.

For each experiment of a GA on a test problem, 100 independent runs with the same set of 100 random seeds to generate initial populations were executed. For each run, the initial population is randomly created using a technique that generates exactly equal number of 0s and 1s for each locus, i.e., $f_1(i,0) = 0.5$ for each locus i in the initial population. In this way the random sampling bias in the initial population (for example for some locus j, $f_1(j,0) = 0.8$ or 0.2) that may mislead SANUM and GBAM_FAD is cancelled.

For each run, the best-so-far fitness was recorded every generation. For each run, the maximum allowable number of generations varies with the test problem and is suitably set to let all GAs have the chance to reach the stable state. Each experimental result was averaged over 100 runs.

5.2 Experimental Results and Analysis

The experimental results on DUFs and 4-Peaks problems are shown in Figure 2 and Figure 3 respectively, from which some results can be observed. First, as recognized by other researchers, the choice of mutation operators and proper probabilities has a significant effect on GA's performance.

Second, on DUF1 and DUF2, the three adaptive mutation schemes outperform traditional mutation regarding the convergence speed toward optimum. This shows that on unimodal functions, the introduction of gene-based adaptive mutation efficiently improves GA's performance. And with the increasing of difficulty level from DUF1 to DUF2, the relative performance of GBAM_FAD improves over GBAM and SANUM. For example, on DUF2 GBAM_FAD achieves the optimum at generation 41, much faster than GBAM and SANUM. This is because from DUF1 to DUF2 the effect of epistasis increases and GBAM_FAD works more efficiently.

Third, on the deceptive function DUF3, only the GA with GBAM_FAD achieved the optimum within 148 generations. All the three other mutation schemes fail to achieve the optimum. And GBAM and SANUM are even beaten by traditional bit flip mutation with respect to GA's performance. For GBAM and SANUM, the existence of deceptive optimum in DUF3 draws the population toward it strongly and once converged GAs with GBAM and SANUM are trapped due to the very low mutation probability for each locus. However, for GBAM_FAD, the correlation scheme of gene-based fitness statistics and gene-based allele distribution stops the population from being trapped by the deceptive optimum efficiently. This is because for each building block in DUF3 when the global and deceptive optima co-exist in the population, the more deceptive optima there are in the population, the more 0-inclined and 1-successful the loci ap-

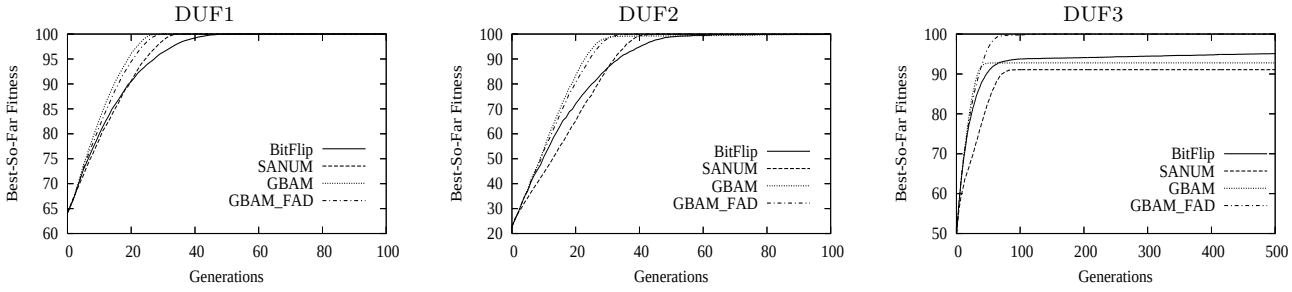

Figure 2: Experimental results of GAs on the DUFs. The data were averaged over 100 runs.

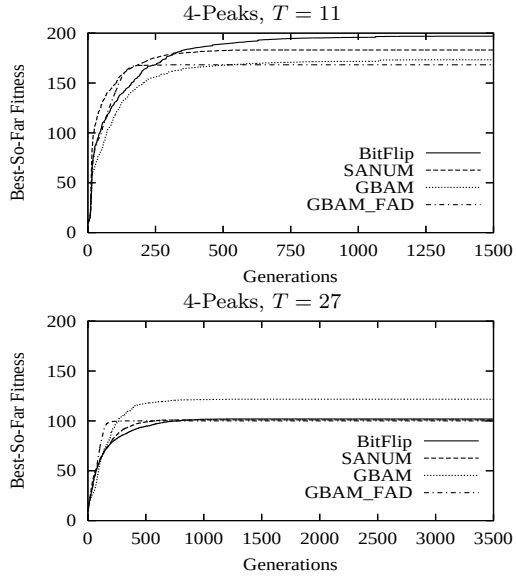

Figure 3: Experimental results of GAs on the 4-Peaks problems.

pear to be. And hence the higher mutation probabilities those loci will have according to Eqs. 4 and 5. This efficiently stops the growth of deceptive optima in the population and leads to the success of GBAM_FAD.

Finally, on the 4-Peaks problems, the performance of GAs drops heavily when $T = 27$ than when $T = 11$. This is as expected since increasing the value of T increases the attraction basin size of the local optima. When $T = 11$, the problem is relatively easy. With relatively bigger mutation probability traditional bit flip mutation can escape the local optima toward the global optimum more easily than adaptive mutation schemes. Hence, bit flip mutation outperforms adaptive mutation schemes. When $T = 27$ all GAs are easily trapped into the local optima and the adaptive mutation schemes converge faster than traditional bit flip mutation.

5.3 Experiments on Convergence Control

From above experimental results it can be seen that the adaptive mutation schemes allow the GAs rapid convergence. On unimodal functions, DUF1 and DUF2, this rapid convergence provides a valuable performance improvement. However, on multi-modal functions (4-Peaks problems) and deceptive function (DUF3) the rapid convergence results in premature and leads the GAs to get stuck at local optima. As addressed in [13], one way to remedy this problem is to implement a mechanism to maintain sufficient diversity in the population in order for the GAs to escape from local optima. In this sub-section we investigate one such mechanism by introducing convergence control, described below.

Every generation the degree of convergence of the population is calculated as the average degree of convergence over all gene loci. For binary encoding, we can derive the degree of convergence for a gene locus from the frequency of 1's in the alleles in that locus. Let $dc(i,t)$ denote the degree of convergence in locus i at generation t and $dc(t)$ the average degree of convergence over all loci at generation t, we have:

$$dc(i,t) = \begin{cases} f_1(i,t), & \text{if } f_1(i,t) > 0.5 \\ 1.0 - f_1(i,t), & \text{otherwise} \end{cases} \quad (8)$$

$$dc(t) = \frac{1}{L}\sum_{i=1}^{i=L} dc(i,t) \quad (9)$$

When $dc(t)$ is higher than a pre-set threshold value T_{dc}, i.e., $dc(t) \geq T_{dc}$, the convergence control strategy takes effect: a predefined portion of $\rho * N$ individuals are randomly selected from the current population, their genotype are complemented, and then the mutation probabilities $p_m^1(i,t)$ and $p_m^0(i,t)$ for GBAM and GBAM_FAD are reset to the initial value, i.e., 0.01 in this study.

In this experimental study we perform convergence control to all GAs with $T_{dc} = 0.95$ and $\rho = 0.2$. The experimental results on DUFs and 4-Peaks problems are shown in Figures 4 and 5 respectively. From these figures, several results can be seen. First, on DUF1 and DUF2 it seems that the convergence control has little effect on the performance of GAs. This is because the complemented individuals are usually discarded by the selection due to their relatively low fitness.

Second, on DUF3 the situation is different. Convergence control improves performance of GBAM and SANUM while it lowers that of GBAM_FAD. This happens because for GBAM and SANUM when the population is converged, a portion of individuals are complemented, which gives them higher chance to escape the deceptive optimum. However, for GBAM_FAD the fitness and allele distribution correlation scheme prevents the GA from being trapped by the deceptive optimum. Hence, when convergence control takes effect, complemented individuals usually have low fitness and may decerase the overall performance if they survive selection. Third, on the 4-Peaks problems, the performance of adaptive mutation schemes are significantly improved by convergence control. For example, GBAM_FAD achieves the optimum fitness of 199 within generation 1309 and generation 3018 for the 4-Peaks problems with $T = 11$ and $T = 27$ respectively for all the 100 runs. This shows that

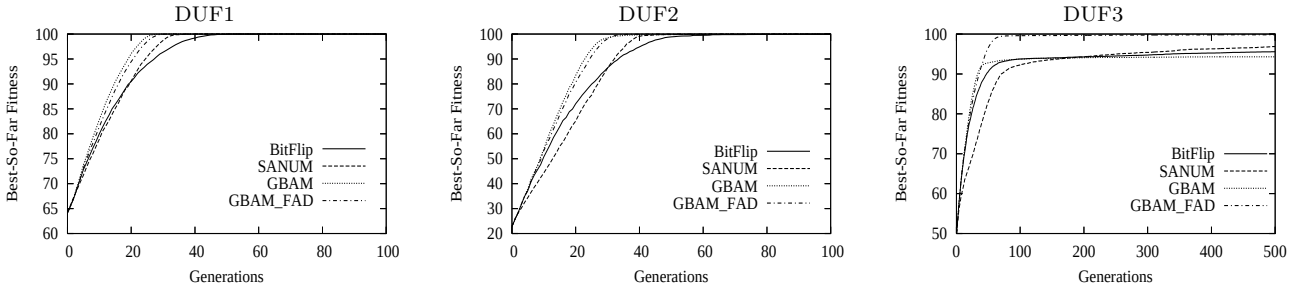

Figure 4: Experimental results of GAs with convergence control on the DUFs.

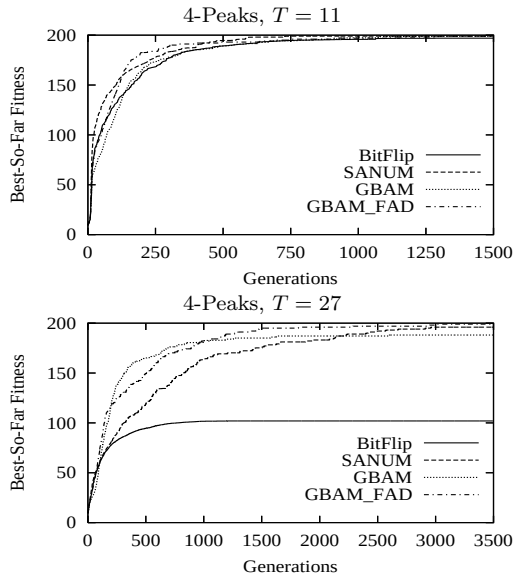

Figure 5: Experimental results of GAs with convergence control on the 4-Peaks problems.

convergence control efficiently improves GA's performance on multi-modal functions.

6. CONCLUSIONS

In this paper, a new gene based adaptive mutation scheme, GBAM_FAD, is proposed for GAs, within which the statistics information on gene based fitness and gene based allele distribution in each gene locus are correlated to explicitly adapt the mutation probability for that locus. Hence the mutation probability for each gene locus is adapted more accurately toward the problem in hand and the evolutionary searching progress. In order to maintain sufficient diversity in the population, a convergence control mechanism is combined with GBAM_FAD. When the population has converged to certain degree a portion of individuals from the population are randomly selected and complemented. The experimental results show that GBAM_FAD efficiently improves GA's performance. Especially, GBAM_FAD efficiently solves the deceptive function, which otherwise is difficult for GAs. Together with the convergence control mechanism, GBAM_FAD also achieves the optimum of difficult 4-Peaks problems, e.g., with $T = 27$. Generally speaking, the experimental results indicate that GAs with GBAM_FAD and the convergence control mechanism are a good choice for difficult optimization problems.

7. REFERENCES

[1] T. Bäck. Self-adaptation in genetic algorithms. In *Proc. of the 1st European Conf. on Artificial Life*, pages 263–271, 1992.

[2] T. Bäck. Mutation parameters. In T. Bäck, D. B. Fogel, and Z. Michalewicz (eds.), *Handbook of Evolutionary Computation*, E1.2.1-E1.2.7, 1997. Oxford University Press.

[3] T. C. Fogarty. Varying the probability of mutation in the genetic algorithm. In *Proc. of the 3rd Int. Conf. on Genetic Algorithms*, pages 104-109, 1989.

[4] D. A. Goldberg. *The Design of Innovation: Lessons from and for Competent Genetic Algorithms*. Boston, MA: Kluwer Academic Publishers, 2002.

[5] J. Hesser and R. Männer. Towards an optimal mutation probability in genetic algorithms. In *Parallel Problem Solving from Nature 1*, pages 23–32, 1991.

[6] B. Julstrom. What have you done for me lately? adapting operator probabilities in a steady-state genetic algorithm. In *Proc. of the 6th Int. Conf. on Genetic Algorithms*, pages 81–87, 1995.

[7] H. Mühlenbein. How genetic algorithms really work: I. mutation and hillclimbing. In *Parallel Problem Solving from Nature 2*, pages 15–29, 1992.

[8] J. D. Schaffer, R. A. Caruana, L. J. Eshelman, and R. Das. A study of control parameters affecting online performance of genetic algorithms for function optimization. In *Proc. of the 3rd Int. Conf. on Genetic Algorithms*, pages 51–60, 1989.

[9] J. E. Smith and T. C. Fogarty. Self-adaptation of mutation rates in a steady-state genetic algorithm. In *Proc. of the 3rd IEEE Conf. on Evolutionary Computation*, pages 318–323, 1996.

[10] W. Spears. Crossover or mutation. In *Foundations of Genetic Algorithms 2*, 1992.

[11] D. Thierens. Adaptive mutation control schemes in genetic algorithms. In *Proc. of the 2002 Congress on Evolutionary Computation*, 2002.

[12] A. Tuson and P. Ross. Adapting operator settings in genetic algorithms. *Evol. Comput.*, **6**(2):161-184, 1998.

[13] S. Uyar, S. Sariel, and G. Eryigit. A gene based adaptive mutation strategy for genetic algorithms. *Proc. of the 2004 Genetic and Evolutionary Computation Conference*, 2004.

[14] S. Yang. Adaptive mutation using statistics mechanism for genetic algorithms. In F. Coenen, A. Preece and A. Macintosh (editors), *Research and Development in Intelligent Systems XX*, pages 19-32, 2003. London: Springer-Verlag.