# A Gene Based Adaptive Mutation Strategy for Genetic Algorithms

Sima Uyar, Sanem Sariel, and Gulsen Eryigit

Istanbul Technical University, Electrical and Electronics Faculty
Department of Computer Engineering, Maslak TR-34469 Istanbul, Turkey
{uyar, sariel, gulsen}@cs.itu.edu.tr

**Abstract.** In this study, a new mechanism that adapts the mutation rate for each locus on the chromosomes, based on feedback obtained from the current population is proposed. Through tests using the one-max problem, it is shown that the proposed scheme improves convergence rate. Further tests are performed using the 4-Peaks and multiple knapsack test problems to compare the performance of the proposed approach with other similar parameter control approaches. A convergence control scheme that provides acceptable performance is chosen to maintain sufficient diversity in the population and implemented for all tested methods to provide fair comparisons. The effects of using a convergence control mechanism are not within the scope of this paper and will be explored in a future study. As a result of the tests, promising results which promote further experimentation are obtained.

## 1 Introduction

Genetic algorithms belong to a class of biologically inspired optimization approaches that model the basic principles of classical Mendelian genetics and Darwinian theory of evolution. Due to their robust nature, genetic algorithms are used in a wide variety of applications. However one of the major drawbacks is that performance largely depends on the appropriate setting of some parameters: population size, crossover and mutation rates. These parameters interact with each other, making it even harder to find optimal settings. However mutation rate is considered to be the most sensitive of these parameters. Mutation has been traditionally regarded as a background operator that mainly works as an insurance policy protecting alleles from being lost from the population. There has been extensive work to investigate the exact nature of mutation and to find optimal settings for different classes of problems [2], [7], [8]. It has also been shown in further studies [2], [3], [12] that using a varying mutation rate strategy overcomes the difficulties of finding optimal mutation rate settings. The techniques developed to set the parameters are classified separately by Eiben et al. [5] Angeline [1] and Smith et al. [10] but the main underlying principles of the different classifications are similar. Parameter setting methods can be classified into two major categories: *parameter tuning* and *parameter control*. In parameter tuning, the parameter values are set in advance, before the run and are kept constant during the whole execution of the algorithm. In parameter control, parameters are initialized at the start of execution and their values are allowed to change during the run. The type

K. Deb et al. (Eds.): GECCO 2004, LNCS 3103, pp. 271–281, 2004.

of the change is defined in [5] to be one of the following: *deterministic* (the parameter value is updated according to some deterministic rule), *adaptive* (the parameter value is updated based on some feedback taken from the population) or *self-adaptive* (the parameter is evaluated and updated by the evolutionary algorithm itself).

In this study, an adaptive mutation rate strategy that increases or decreases the mutation rate for each locus on the chromosome, based on feedback obtained from the current population is introduced. Even though using feedback from the current state of the search seems to be a useful approach, it has not been studied much within the scope of canonical genetic algorithms [12]. This approach is tested against previously published methods for mutation rate control on a chosen set of test problems. The results are seen to be promising and promote further study. The rest of this paper is organized as follows: Section 2 introduces the proposed mutation rate adaptation approach section 3 presents the experimental setup, section 4 discusses the results of the experiments, and section 5 provides a conclusion and possible directions for future work.

## 2 A Genetic Algorithm with Gene Based Adaptive Mutation

Mutation as an insurance policy against permanent loss of genes is considered to be the most sensitive of the required GA parameters. Determining the optimum fixed mutation rate for different types of problems requires an empirical analysis. In this paper, a Gene Based Adaptive Mutation (GBAM) method is proposed. This approach experiments with adaptive mutation rate values during the run using feedback from the population. Therefore, instead of using a fixed optimum value for a mutation rate, a range can be specified which provides more flexibility. Different from other known mutation adaptation strategies, GBAM has its own mutation rate value for each locus. An adaptive approach for adjusting mutation rates for the gene locations based on the feedback obtained by observing the relative success or failure of the individuals in the population is used. Since the mutation rates at each locus depend mainly on whether the individuals with a specific allele value for that locus is successful or not, GBAM is more suited to problems in which the representation is binary.

In GBAM, there are two different mutation rates defined for each locus: $p_{m1}$ for those genes that are "1" and $p_{m0}$ for those that are "0". In the reproduction phase, the appropriate mutation rate is applied based on the gene allele value. Initially all of the mutation rates are set to an initial value in the specified boundaries. Then for each generation, the mutation probabilities $p_{m1}$ and $p_{m0}$ for each locus are updated based on feedback taken from the relative success or failures of those individuals having a "1" or "0" at that locus. The update rule for the two mutation rate values for one gene location can be seen in Eq.1. This update rule is applied separately for each locus. The $p_{mi}$ value for a locus corresponds to the rate of mutation, which will be applied when the gene value is $i$ in the corresponding gene location. $S_{avg}$ is the average fitness of the individuals with an allele “1” for the corresponding gene location. $P_{avg}$ is the average fitness of the population, $\gamma$ is the update value for the mutation rates. For a maximization problem, if the ratio of $S_{avg}$ to $P_{avg}$ is greater than 1 or all the genes at that locus are 1, i.e. the average fitness value of the individuals having the allele "1" for that locus is higher than those having "0", the allele value “1” for the corresponding locus is assumed to generate more successful results. Therefore, a decrease in $p_{m1}$, and

an increase in $p_{m0}$ for the corresponding locus are implemented. Similarly, if the $S_{avg}$ to $P_{avg}$ ratio is less than 1 or all the genes at that locus are 0, then the opposite operations are implemented on the mutation rate values. In the case of a minimization problem, the operations in Eq. 1 should be exchanged. As a result of the updates at each generation, $p_{mi}$ values are allowed to oscillate within the limits defined by lower and upper bounds. If an update causes a mutation rate to exceed the limits, it is set to the corresponding boundary value. All GBAM parameters are determined empirically.

$$p_m^+ = \begin{cases} p_{m1}^+ = p_{m1} - \gamma \quad \wedge \quad p_{m0}^+ = p_{m0} + \gamma, \quad (S_{avg} / P_{avg}) \geq 1 \vee \forall\, gene = 1 \\ p_{m1}^+ = p_{m1} + \gamma \quad \wedge \quad p_{m0}^+ = p_{m0} - \gamma, \quad (S_{avg} / P_{avg}) < 1 \vee \forall\, gene = 0 \end{cases} \quad (1)$$

As will be shown in the analysis of the experiments, GBAM allows rapid convergence. For unimodal objective functions, this rapid convergence provides a valuable refinement. However, the premature convergence problem, which may cause the program to get stuck at local optima, arises especially for the multimodal objective functions. This problem is explored in detail in [9] for self-adaptive mutations, however the results can easily be extended to adaptive mutation schemes too. One way to remedy this is to implement a mechanism to maintain sufficient diversity in the population for escaping local optima. As a result of preliminary experimentation, a method that complements ($p_{mi}$=1.0) the genotype of a predefined percentage of the population during the reproduction phase when the population converges, and then resets the mutation rates to their initial values is observed to generate acceptable results. This convergence control method is used in GBAM to enable the program to escape from possible local optima.

## 3 Experimental Design

The aim of the experiments is to show that GBAM requires fewer generations to reach the optimum as well as exploring its performance compared to other parameter control approaches found in literature, based on two different types of problems. The testing phase consists of two stages. In the first stage, GBAM is compared with a simple canonical genetic algorithm using a fixed mutation rate to determine whether the addition of the proposed adaptive mutation rate strategy causes a performance improvement by reducing the amount of fitness evaluations to reach the optimum. There is no convergence control used at this stage. The one-max problem, which is unimodal and easy for the simple genetic algorithm, is used for this stage of the tests. In the second stage of the testing phase, representative parameter control approaches for each type of change scheme (deterministic, adaptive and self-adaptive) developed for canonical genetic algorithms are chosen from literature and are compared with GBAM. Two test problems are used during this stage: 4-Peaks and the multiple knapsack problems. For both of these problems, all tested approaches are equipped with the convergence control mechanism explained in Section 2 to provide fair comparisons. The results are evaluated based on their solution quality and the number of fitness evaluations it takes each approach to find those results.

### 3.1 Test Problems

**One-Max:** The main aim of this problem is to maximize the number of 1s in a binary represented string of length L. The optimum for this function is L.

**4-Peaks:** The fitness function for the 4-Peaks problem where each individual consists of 100 bits is given in Eq.2 where z(x) is the number of contiguous 0s ending in Position 100, o(x) is the number of contiguous 1s starting in Position 1, and T is a threshold. The problem has two global and two local optima. As explained in [4], by increasing T, the basins of attraction surrounding the inferior local optima increase in size exponentially while the basins around the global optima decrease at the same rate. Therefore, increasing T makes it harder for a GA to escape the local optima.

$$REWARD = \begin{cases} 100+T & if \quad o(x) > T \wedge z(x) > T \\ 0 & otherwise \end{cases}$$
$$f(x) = MAX(o(x), z(x)) + REWARD \tag{2}$$

**Multiple Knapsack Problem:** In the 0/1 Multiple Knapsack Problem (Mkp), there are $m$ knapsacks of capacity $c_j$, $n$ objects of profit $p_i$. The weights for the objects are different for each knapsack. $w_{ij}$ represents the $i^{th}$ object's weight for the $j^{th}$ knapsack. A feasible vector solution for this problem can be defined as a vector $\vec{x} = (x_1, x_2, ..., x_n)$ where $x_i \in \{0,1\}$, such that $\sum_{i=1}^{n} w_{ij} * x_i \leq c_j$ for $j = 1,2,..m$. The value "0" for an object in the vector representation means that the object is not placed in any of the knapsacks. Otherwise, the object is placed in all of the knapsacks. The main objective is to find a feasible vector with maximum profit $P(\vec{x}) = \sum_{i=1}^{n} x_i * p_i$. The feasible vector solution should satisfy the constraint that no knapsack is overfilled. A penalty value is added to the objective function to enable the feasible individuals to have more survivability. The penalized objective function of [6] defined in Eq. 3 is used in this study.

$$f(\vec{x}) = \sum_{i=1}^{n} p_i * x_i - \frac{(\max\{p_i\}+1)}{\min\{w_{ij}\}} * \max_{j=1,2..m} (\max(0, \sum_{i=1}^{n} w_{ij} * x_i - c_j)) \tag{3}$$

### 3.2 Parameter Control Approaches Chosen for Comparisons

There are different formulations and implementations of various parameter control techniques in literature. A representative scheme that is shown to give good performance is chosen from each category and used for the comparisons.

**Deterministic Approach:** Deterministic mutation rate schedule provides the mutation rate to be deterministically altered at each generation. The mutation rate decreases from a value (generally 0.5) to the optimum mutation rate (generally 1/L) without using any feedback from the population. The deterministic mutation rate schedule suggested in [10] was reported in [12] as being successful for hard combinatorial problems. Time-varying mutation rate $p_t$ is calculated based on the

formula given in Eq. 4. In this formula, $t \in \{0,1,...,T-1\}$ denotes the generation number, and T is the maximum number of generations.

$$p_t = \left(2 + \frac{L-2}{T-1} * t\right)^{-1} \tag{4}$$

**Self-Adaptive Approach:** In the self-adaptive approach, the parameters are encoded into the chromosomes and undergo mutation and recombination. The idea is that better parameter values lead to better individuals and these parameter values will survive in the population since they are brought together with the surviving individuals. In [3] a self-adaptation mechanism of a single mutation rate per individual is proposed. The mutation of this mutation rate p ∈ ]0,1[ gives the new mutation rate p' ∈ ]0,1[ according to Eq. 5. In this equation γ is the learning rate which controls the adaptation speed and it is taken as 0.22 in [3]. An individual consists of a bit string and an individual mutation rate p. The new individual is determined through bit wise mutation of n bits using the mutated mutation rate value p'. The mutation rate is not allowed to go below 1/L. In this approach the crossover is applied only to the binary vector and has no effect on p.

$$p' = (1 + \frac{1-p}{p}.\exp(-\gamma.N(0,1)))^{-1} \tag{5}$$

**Adaptive Approach:** Adaptive GA proposed in [11] is a kind of individually adaptive mutation rate strategy. The probabilities of crossover and mutation are adapted depending on the fitness values of the individuals. The adaptation of the $p_c$ and $p_m$ allows the individuals having fitness values of over-average to maintain their genetic material, while forcing the individuals with sub-average fitness values to disrupt. Therefore, the method auto-controls the convergence situation. The method is tested with only SGA in [11]. In Adaptive GA both the mutation and the crossover rates are adapted. However, since the effects of the crossover rate adaptation are not addressed in this study, only the mutation rate adaptation strategy of the Adaptive GA is used as a comparison method. The mutation rate adaptation rule is given in Eq. 6. In this equation, $f$ denotes the fitness value of the individual, $f_{\max}$ denotes the best fitness value of the current generation, and $f_{avg}$ denotes the average fitness value of the current generation. In [11], the constants $k_2$ and the $k_4$ are chosen as 0.5.

$$\begin{aligned} p_m &= k_2(f_{\max} - f)/(f_{\max} - f_{avg}), \quad f \geq \bar{f} \\ p_m &= k_4 \quad f < f_{avg} \end{aligned} \tag{6}$$

## 4 Experimental Results

As explained in the previous section, the experiments consist of two stages. For all the tests in each stage, the program implementation for each chosen approach on each test problem is run 100 times. In this section the results of each stage will be given separately. In tables, μ denotes mean values, σ standard deviations and the 99% CI

confidence intervals. Some parameter settings are kept constant through all tests. A population consists of 250 individuals. Parent selection is done through tournament selection with tournament sizes of two. Recombination is done through two-point cross over at a fixed rate of 1.0. The new population is determined using elitism where the best individual replaces the worst individual in the next generation.

### 4.1 Exploring the Effects of GBAM on Number of Generations to Find Optimum

The GBAM approach is expected to reduce the number of generations to locate an optimal individual. To investigate just the effect of the proposed mutation rate adaptation approach, GBAM and a simple canonical genetic algorithm (SGA) is applied to the One-Max problem. Since the problem space is unimodal, no convergence control is implemented. Maximum number of generations for both GBAM and SGA are 500. The mutation rate for SGA is 1/L, where L is the string length. For GBAM, the initial mutation rate is 0.02 and this mutation rate value is allowed to change between a lower bound of 0.0001 and an upper bound of 0.2 with update value $\gamma$=0.001 in Eq.1. These settings are determined empirically to provide the best performance for each approach. Tests are performed for three different string lengths: L=200, L=400, L=800. Both GBAM and SGA are able to locate the optimum individuals for all tested string lengths. The statistical calculations are given in Table1. Here 99%CI shows the 99% confidence interval of the difference between the means of GBAM and SGA. Based on the results in Table-1, it can be said with 99% certainty that the true mean value for the reduction in number of generations to reach the optimum when using GBAM for the one-max problem with the chosen parameter settings lies within the given CI ranges. This result confirms the expectation that GBAM locates the optimal individual much quicker than a SGA under similar circumstances.

**Table 1.** Statistical calculations for number of generations to reach the optimum

| | L=200 | | L=400 | | L=800 | |
|---|---|---|---|---|---|---|
| | $\mu$ | $\sigma$ | $\mu$ | $\sigma$ | $\mu$ | $\sigma$ |
| **GBAM** | 48.09 | 2.56 | 99.28 | 34.27 | 217.33 | 39.23 |
| **SGA** | 91.08 | 7.33 | 169.65 | 14.04 | 332.19 | 23.18 |
| **99%CI** | 41.45 to 44.53 | | 62.97 to 79.97 | | 103.1 to 126.64 | |

### 4.2 Comparison of GBAM with Other Parameter Control Methods

The aim of this second testing stage is to compare the performance of GBAM with the different parameter control approaches explained in section 3.2. Two kinds of test problems are used during these tests: 4-Peaks (section 3.1.2) and Mkp (section 3.1.3). For 4-Peaks problem, the methods given in [4] are tested for different T values between 11 and 25 and it is seen that after the value of 19 it becomes hard for simple GAs to find the optimum. Therefore in this study T=11 is chosen to test an easy 4-

Peaks problem and T=27 a difficult one. These values are chosen to compare the effectiveness of the algorithms for different levels of difficulty. The global best fitness value is 199, while local best fitness value is 100. The number of maximum generations is selected as 3500 for this problem. As a testbed for the Mkp Weing-7 and Weish-30 datasets [13] are selected. In Weing-7, there are 2 knapsacks and 105 objects. In Weish-30, there are 5 knapsacks and 90 objects. The known optima are reported as 1095445 for Weing-7 and as 11191 for Weish-30 in [13]. The number of maximum generations is selected as 6000. Because the chosen penalty approach assigns high negative fitness values to infeasible individuals, for the Mkp instances, the mutation rate adaptation is done based on only the feasible individuals.

A convergence control mechanism is implemented for each of the algorithms except the Adaptive GA which has its own convergence control. The implemented mechanism takes the complement of the 25% of the population when 90% of all the genes are converged for all gene locations. Chromosome length (L) is equal to 100 for 4-Peaks, and to the number of objects for the Mkp instances. The initial mutation rate for SA is 2/L. In GBAM, the initial mutation rate is 1/L for all problem instances. The upper and lower bounds for the mutation rate in GBAM are 0.2 and 0.0001 respectively, with an update amount of $\gamma$=0.001. The comparison criteria for the tests are chosen as the best fitness values and the number of generations to reach the best fitness both averaged over 100 runs. The following abbreviations are used in Table-2 and Table-3: SA (Self Adaptive), Adap (Adaptive), Det (Deterministic) and GBAM.

**Table 2.** Statistical results of 4-Peak Problem Instances

| | | 4-Peaks (T=11) | | | 4-Peaks (T=27) | | |
|---|---|---|---|---|---|---|---|
| BEST FITNESS | | $\mu$ | $\sigma$ | 99%CI | $\mu$ | $\sigma$ | 99%CI |
| | **SA** | 143.94 | 7.98 | 141.80-146.08 | 43.44 | 8.24 | 41.23-45.65 |
| | **Adap** | 128.98 | 46.18 | 116.60-141.36 | 95.53 | 3.53 | 94.58-96.48 |
| | **Det** | 198.83 | 0.64 | 198.66-199.00 | 99.82 | 0.88 | 99.67-99.97 |
| | **GBAM** | 199.00 | 0.00 | 199.00-199.00 | 199.00 | 0.00 | 199.00-199.00 |
| NO OF GENS | | 4-Peaks (T=11) | | | 4-Peaks (T=27) | | |
| | | $\mu$ | $\sigma$ | 99%CI | $\mu$ | $\sigma$ | 99%CI |
| | **SA** | 2481.02 | 801.55 | 2266.19-2695.85 | 2302.51 | 852.95 | 2073.90-2531.12 |
| | **Adap** | 3020.88 | 404.14 | 2912.57-3129.20 | 2962.98 | 403.61 | 2854.81-3071.15 |
| | **Det** | 3142.52 | 227.71 | 3081.49-3203.55 | 3099.12 | 293.37 | 3020.49-3177.75 |
| | **GBAM** | 301.85 | 209.47 | 245.71-357.99 | 654.07 | 466.40 | 529.07-779.07 |

Table-2 and Table-3 are given in two parts: one is for the best fitness values and the other is for the number of generations to locate the individual with the best fitness. To assess the success of an approach, both parts should be considered together. Based on the results in these tables, GBAM seems to be promising for all of the tested problems. For the 4-Peaks problem, GBAM finds the global optimum for all of the T values, while none of the other methods can. The confidence intervals also do not

**Table 3.** Statistical results of MKP Problem Instances

| | | MKP (Weing-7) | | | MKP (Weish-30) | | |
|---|---|---|---|---|---|---|---|
| | | μ | σ | 99%CI | μ | σ | 99%CI |
| BEST FITNESS | **SA** | 1051736.88 | 15499.66 | 1047582.69-1055891.00 | 10030.18 | 381.97 | 9927.81-10132.55 |
| | **Adap** | 1083460.50 | 6146.87 | 1081813.00-1085108.00 | 10860.07 | 144.43 | 10821.36-10898.78 |
| | **Det** | 1094818.63 | 462.55 | 1094694.63-1094942.63 | 11163.94 | 16.18 | 11159.60-11168.28 |
| | **GBAM** | 1095085.25 | 604.02 | 1094923.38-1095247.13 | 11190.96 | 0.40 | 11190.85-11191.07 |
| | | MKP (Weing-7) | | | MKP (Weish-30) | | |
| | | μ | σ | 99%CI | μ | σ | 99%CI |
| NO OF GENS | **SA** | 3520.13 | 1678.82 | 3070.18-3970.08 | 3527.80 | 1746.44 | 3059.73-3995.87 |
| | **Adap** | 5040.67 | 858.49 | 4810.58-5270.76 | 4739.13 | 1100.37 | 4444.21-5034.05 |
| | **Det** | 5484.20 | 430.95 | 5368.70-5599.70 | 5461.25 | 521.09 | 5321.59-5600.91 |
| | **GBAM** | 2586.80 | 1788.95 | 2107.33-3066.27 | 588.86 | 695.30 | 402.51-775.21 |

intersect, showing that in 99% of all trials, the mean values for all approaches will fall within these intervals, confirming that GBAM performs better in these cases. For Mkp, GBAM generates most successful results of all, both in the average fitness value and the number of generations needed to find the best fitness. The 99% confidence intervals for the average fitness values for the Weing-7 instance intersect for Det and GBAM, however it should be noted that the number of generations for GBAM to reach its best fitness value is much less than that for Det.

The best fitness values averaged over 100 runs for all generations can be seen in Fig 1 for the tested problems. The deterministic approach behaves as expected with increasing best fitness values for each generation. The convergence intervention applied to this algorithm is not effective due to the nature of the approach since convergence occurs towards the end of the run. However in GBAM convergence can occur in earlier steps, because this method forces the population to converge fast. In GBAM after the population converges and 25% of the population is complemented, the mutation rate values for the genes are again reset to the initial values. However, due to elitism, previously found good individuals are not lost. In GBAM the overall best individual fitness values are highest of all approaches, and they are found in earlier generations as mentioned before. The best fitness values found for the Mkp which are higher than 1095000 for Weing-7 (listed as succesful in [12]) and 11150 for Weish-30 can be seen in Table-4. Based on the graphical results in Fig. 1, the rise of the plot for GBAM is fastest of all the tested methods.

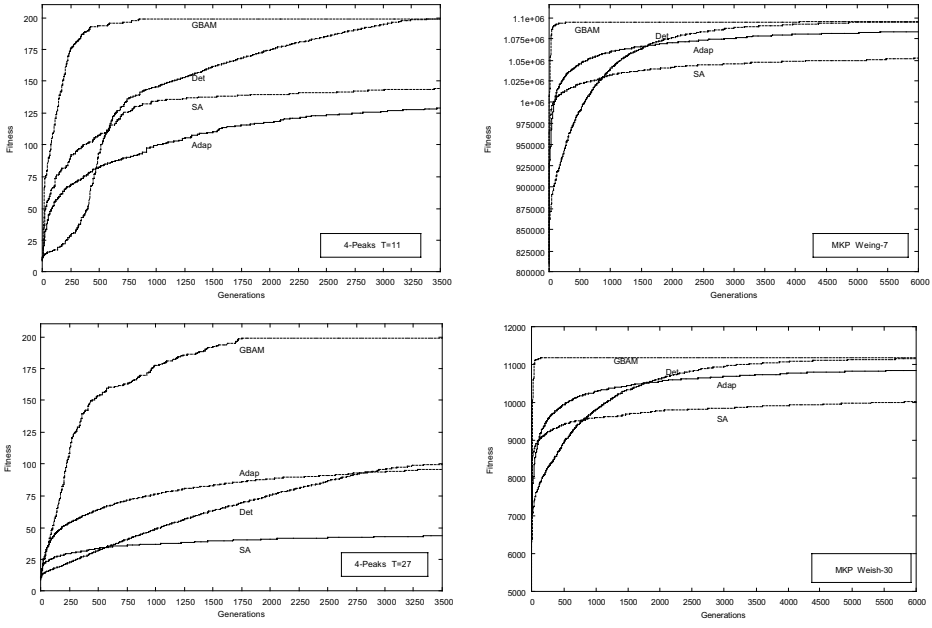


**Fig. 1.** Avg. best fitnesses over 100 runs

## 5 Conclusion and Future Work

In this study, a mutation rate adaptation approach (GBAM) for each gene location in a binary representation, based on the relative performance of individuals is proposed. GBAM is expected to increase the convergence rate. This has been confirmed through tests performed on a simple one-max problem. However, especially in multimodal problems, fast convergence may cause the population to get stuck on local optima. To overcome this problem, a convergence control mechanism that introduces diversity when the population converges, is added. This convergence control is applied to all tested algorithms but since GBAM again adapts the mutation rate parameter, it quickly converges after the introduction of diversity and thus is able to locate good solutions in fewer generations even for multimodal problems. This fact has been shown through tests using two instances of both the multiple knapsack problem and a problem with a fitness landscape that has local optima with large basins of attraction. In both cases GBAM is shown statistically to perform better than the other tested methods both in locating the optima and also in reaching the optima in fewer generations. One major drawback is that GBAM increases the computational costs since at every generation new mutation rate values for all gene locations (as many as the length of a chromosome) are recalculated. On the other hand, among the other tested methods, the deterministic approach seems to be able to reach acceptable fitness values with almost no additional computational costs, however it takes much

**Table 4.** Best fitness values higher than selected thresholds observed in 100 runs for MKP (1095000 for Weish-30 and 11150 for Weing-7)

| Weing-7 Results | | | | |
|---|---|---|---|---|
| | SA | Adap | DET | GBAM |
| 1095445 | - | - | - | 10 |
| 1095382 | - | - | 5 | 18 |
| 1095357 | - | - | 3 | 12 |
| 1095295 | - | - | - | 1 |
| 1095266 | - | - | - | 6 |
| 1095264 | - | - | 3 | 3 |
| 1095262 | - | - | - | 2 |
| 1095232 | - | - | 1 | 1 |
| 1095207 | - | - | 2 | 5 |
| 1095206 | - | - | 4 | 2 |
| 1095202 | - | - | 1 | - |
| 1095195 | - | - | 1 | - |
| 1095157 | - | - | 1 | 3 |
| 1095141 | - | - | 1 | 1 |
| 1095137 | - | - | 3 | 4 |
| 1095132 | - | - | - | 1 |
| 1095114 | - | - | 1 | 1 |
| 1095112 | - | - | 3 | 2 |
| 1095081 | - | - | 2 | - |
| 1095065 | - | - | - | 2 |
| 1095062 | - | - | 3 | 1 |
| 1095057 | - | - | - | 1 |
| 1095056 | - | - | - | 1 |
| 1095039 | - | - | 1 | - |
| 1095035 | - | - | 1 | 3 |
| 1095014 | - | - | 1 | - |
| 1095007 | - | - | 2 | - |
| **Overall** | 0/100 | 0/100 | 39/100 | 80/100 |

| Weish-30 Results | | | | |
|---|---|---|---|---|
| | SA | Adap | DET | GBAM |
| 11191 | - | - | 11 | 99 |
| 11187 | - | - | 5 | 1 |
| 11182 | - | - | 2 | - |
| 11181 | - | - | 1 | - |
| 11178 | - | - | 1 | - |
| 11177 | - | - | 1 | - |
| 11175 | - | - | 1 | - |
| 11174 | - | - | 2 | - |
| 11173 | - | - | 4 | - |
| 11172 | - | - | 2 | - |
| 11170 | - | - | 4 | - |
| 11169 | - | - | 4 | - |
| 11168 | - | - | 3 | - |
| 11167 | - | - | 5 | - |
| 11166 | - | - | 1 | - |
| 11165 | - | - | 1 | - |
| 11164 | - | - | 1 | - |
| 11162 | - | - | 3 | - |
| 11161 | - | - | 2 | - |
| 11160 | - | - | 5 | - |
| 11159 | - | - | 1 | - |
| 11158 | - | - | 3 | - |
| 11157 | - | - | 5 | - |
| 11156 | - | - | 1 | - |
| 11155 | - | - | 1 | - |
| 11154 | - | - | 1 | - |
| 11153 | - | - | 6 | - |
| 11152 | - | 1 | - | - |
| 11151 | - | - | 5 | - |
| **Overall** | 0/100 | 1/100 | 82/100 | 100/100 |

longer. So for problems similar to the ones used in this study, in the cases where it is more important to find good results in fewer generations, GBAM seems to be the better choice among the tested methods.

Even though as a result of these preliminary tests, the overall performance of GBAM seems to be very promising for the chosen type of problems, there is still more work to be done to be able to make healthy generalizations. First of all, the parameter settings for GBAM, such as the lower and upper bound values, initial mutation rate and the mutation update value have been determined experimentally. More experiments need to be performed to see the effects of these parameters on performance more thoroughly. Secondly, the test problem set can be extended to include different types of problem domains. Thirdly, a convergence control

mechanism has been implemented for this study, however its effects have not been thoroughly examined. More rigorous experimentation needs to be performed to be able to fully understand the exact nature of the convergence control method. Finally, it is observed that the adaptive nature of the proposed mutation mechanism might be suitable to be used in dynamic environments. Since the mutation rate adapts itself based on the relative fitness of individuals, it would also be able to adapt to a change in the environment causing a change in the fitness values of the individuals and it would not need to explicitly detect the change in the environment. Based on these observations, it seems to be a promising line of further study to use GBAM in dynamic environments in its pure form or even in combination with other approaches developed previously to cope with changing environments.

**Acknowledgement.** The authors would like to thank Jens Gottlieb for his helpful suggestions regarding the penalty function for MKP.

## References

1. Angeline P. J.: Adaptive and Self-adaptive Evolutionary Computation. Computational Intelligence, A Dynamic System Perspective, IEEE (1995) 152-161
2. Bäck T.: Optimal Mutation Rates in Genetic Search. Proc. of 5th International Conference on Genetic Algorithms, Morgan Kaufmann (1993)
3. Bäck T., Schlütz M.: Intelligent Mutation Rate Control in Canonical Genetic Algorithms. Proc. Int. Symp. on Methodologies for Intelligent Syst. (1996) 158-167
4. Baluja S., Caruana. R.: Removing the Genetics from the Standard Genetic Algorithm. Proc. 12. Int. Conf. on Machine Learning, Morgan Kaufmann, (1995) 38-46
5. Eiben A. E., Hinterding R., Michalewicz Z.: Parameter Control in Evolutionary Algorithms. IEEE Trans. on Evolutionary Computation, Vol. 3, No.2. (1999) 124-141
6. Gottlieb J.: On the Feasibility Problem of Penalty-Based Evolutionary Algorithms for Knapsack Problems. Proc. of EvoWorkshops, Lecture Notes in Computer Science Vol. 2037, Springer (2001) 50-59
7. Hinterding R., Gielewski H., Peachey T. C.: The Nature of Mutation in Genetic Algorithms. Proc. 6. Int. Conf. on GAs, Morgan Kaufmann (1995) 65-72
8. Ochoa G.: Setting the Mutation Rate: Scope and Limitations of the 1/L Heuristic. Proc. Genetic and Evolutionary Comp. Conf., Morgan Kaufmann (2002)
9. Rudolph G.: Self-Adaptive Mutations May Lead to Premature Convergence. IEEE Trans. on Evolutionary Computation, Vol. 5., No. 4. (2001) 410-414
10. Smith J. E., Fogarty T. C.: Operator and Parameter Adaptation in Genetic Algorithms. Soft Computing 1, Springer-Verlag (1997) 81-87
11. Srinivas, M., Patnaik, L. M.: Adaptive Probabilities of Crossover and Mutation in Genetic Algorithms. IEEE Trans. on Systems, Man and Cybernetics, Vol. 24. No. 4. (1994) 656-667
12. Thierens D.: Adaptive Mutation Control Schemes in Genetic Algorithms. Proc. of Congress on Evolutionary Computing, IEEE (2002)
13. weing7.dat, weish30.dat: http://elib.zib.de/pub/Packages/mp-testdata/ip/sac94-suite/