Journal of Naval Science and Engineering
2009, Vol. 5 , No.3, pp. 26-37

EVALUATION OF CANONICAL ALGORITHMS AND ADAPTIVE ALGORITHMS FOR TWO KNOWN PROBLEMS

Asst. Prof. Müjdat SOYTÜRK, LCDR
Turkish Naval Academy
Naval Science and Engineering Institute
Tuzla, Istanbul, Turkiye
msoyturk@dho.edu.tr

Abstract

In recent year years, adaptive approaches are getting more interest in application areas. On the other hand, canonical algorithms keep their importance as a first step solution approach and for comparison with adaptive approaches. In this paper, two problems, namely the One-Max Problem and the Generalized Rastrigin's Function, are solved using generational canonical algorithms with fixed mutation rate parameter and self-adaptive mutation rate parameter. For these problems, solution results of self-adaptive methods are compared with the results of deterministic methods. Observed results provide interesting results for these problems.

KANONİK ALGORİTMALAR VE UYARLANABİLİR ALGORİTMALARIN BİLİNEN İKİ PROBLEM İÇİN DEĞERLENDİRİLMESİ

Özetçe

Son yıllarda uyarlanabilir yaklaşımlar uygulama alanlarında daha fazla ilgi görmektedir. Diğer taraftan, başvurulan ilk çözüm yöntemi olması ve uyarlanabilir algoritmaların karşılaştırılmasında kullanılması nedeniyle, kanonik algoritmalar hala önemlerini korumaktadırlar. Bu makalede, One-Max Problemi ve Genelleştirilmiş Rastrigin's Fonksiyonu, hem sabit *mutasyon oranı hem de kendinden-uyarlamalı mutasyon oranı kullanılarak çözülmüştür. Kendinden uyarlamalı yöntem ile elde edilen sonuçlar, belirleyici yöntemden elde edilen sonuçlar ile karşılaştırılmıştır. Sonuçların, değerli katkısı olmuştur.*

Keywords: *Canonical Algorithms, Adaptive Algorithms, One-Max Problem, Rastrigin's Function, Genetic Algorithms, Evolutionary algorithms.*
Anahtar Kelimeler: *Kanonik Algoritmalar, Uyarlanabilir Algoritmalar, One-Max Problemi, Rastrigin's Fonksiyonu, Genetik Algoritmalar, Gelişimsel Algoritmalar.*

1. INTRODUCTION

Evolutionary algorithms are composed of two processes: "(1) selection, which differentially boosts the frequency in the population of those forms favored by the fitness function, and (2) variation, which then stochastically perturbs the selected forms, hopefully yielding a few of higher fitness than any previously found" [4]. The performance of evolutionary algorithms is thus ultimately limited by the ability of the variation process to continue to generate new forms of higher fitness. Therefore, we can use the self-adaptation to provide variation. The most common class of self-adaptation is the mutation rate.

The objective of this paper is to observe the effects of the varying mutation to the mean fitness values and to the selection pressure. Therefore, we implemented some computer programs to solve the one-max problem by a generational canonical genetic algorithm (GA) with fixed mutation rate and self-adaptive mutation rate, and write another computer program to solve the Generalized Rastrigin's Function. Observed results are compared to make conclusion.

In the second section, we describe the given problem and propose better solutions to them. We describe the simulation environment and parameters in the third section. In the forth section, we give the results of the experiments. We conclude the paper in the last-fifth-section.

2. PROBLEM DEFINITION

The fitness function of the one-max problem is:

$$f(x) = \sum_{i=1}^{L} x_i$$

where L is the length of the chromosome, and x_i is the genes.

The objective of the one-max problem is to get the maximum fitness value.

The generalized Rastrigin's function is:

$$f\left(\bar{x}\right) = n * A + \sum_{i=1}^{n} x_i^2 - A * \cos\left(w * x_i\right)$$

where $A=10$, $w = 2\pi$, $-5.12 <= x_i < 5.12$.

The objective of the Rastrigin's function is to obtain minimum fitness value with obtaining 0 (zero).

We implement these two problems with deterministic parameter control and self-adaptive parameter control on mutation rates. The deterministic parameter control takes place when the value of a strategy is altered by some deterministic rule. This rule modifies the strategy parameter deterministically without using any feedback from the search. In the given problem, this is the fixed mutation rate.

Rather than including mutation rates among the global parameters that must be set in implementations of artificial evolution, an alternative is to encode one or more mutation rates along with each individual in the population. When the mutation operator is applied to an individual, the mutational events then occur with a probability determined by the individual's own encoded mutation rate(s), rather than a globally-fixed parameter. The encoded mutation rates themselves are also subject to mutation, so variation among the rates themselves is continuously maintained in the population, Selection will then favor some rates of others to the extent to which particular rates are more often associated with individuals of high fitness [1].

The magnitude of the mutation of the mutation rates themselves is determined by another parameter γ, equal to 0.22, which is equal to 0.22 and refers to the standard deviation of an exponential Gaussian distribution from which a value is sampled. The product of this value and the mutation rate is then becomes the new mutation rate. This form of mutation-rate-mutation varies in proportion to their own size [1].

In the self-adaptive approach, control parameters changes according to the online feedback. The parameters to be adopted are encoded into the chromosomes and undergo mutation and recombination. The better values of these encoded parameters lead to better individuals. In the given problem, the self-adaptive mutation rate is the only control parameter. The idea is that better mutation rates will produce better off-spring and then hitchhike on their improved children to new generations, while bad rates will die out [3].

3. SIMULATIONS

We evaluated both of the algorithms with simulations. We implemented the simulations in MS Visual C++ editor with C programming language and compiled with MS Visual C++ compiler. We made 1000 runs for One-Max Problem simulation and 100 runs for Generalized Rastringin's Function simulation. The results are tested with some statistical tests in MS Excel.

3.1 Simulation Parameters

In all simulations, we use some constant parameter values. The parameter settings are given below:

Representation	binary strings (gray code is used for the Rastrigin's function.
Initialization	random
Parent Selection	tournament selection with k=2
Crossover	two point with probability 0.01
Mutation	bit-flip mutation with probability 0.01 with fixed mutation rate.
Replacement	generational without elitism
Population Size	250
Termination	After 3000 generations of after optimum found

Table 1: Parameter Settings

In the programs that use self-adaptive mutation rate, the mutation rate changes during the simulation. It is set to a uniform random value *U[0,1]* at the beginning of the simulation not smaller than *1/n*, *n* is the chromosome length. During the simulation, at each generation and at each chromosome, the mutation rate changes according to values from normal distribution *N[0,1]*. Changed mutation affects the current individual to produce new mutated individual for the next generation. A γ value, equal to 0.22, is taken for determining mutation rate as described in [1].

3.2 Performance Metrics

In the Generalized Rastrigin's Function, we observe the average fitness and best fitness values with the (time table) generation numbers. We also observe the average fitness values for the One-Max problem. There is a correlation between an individual's mutation rate and the expected number of offspring, because they are assigned via the selection process. At any given generation, those individuals favored by the selection process–the highest fitness individuals. Therefore, we observe the changes of average mutation rates with respect to fitness values in self-adapted tests.

4. SIMULATIONS

Best and average fitness results are shown in Figure 1 and Figure 2. In Figure 3 and Figure 4, each 50 values averaged to make the figures clearer.

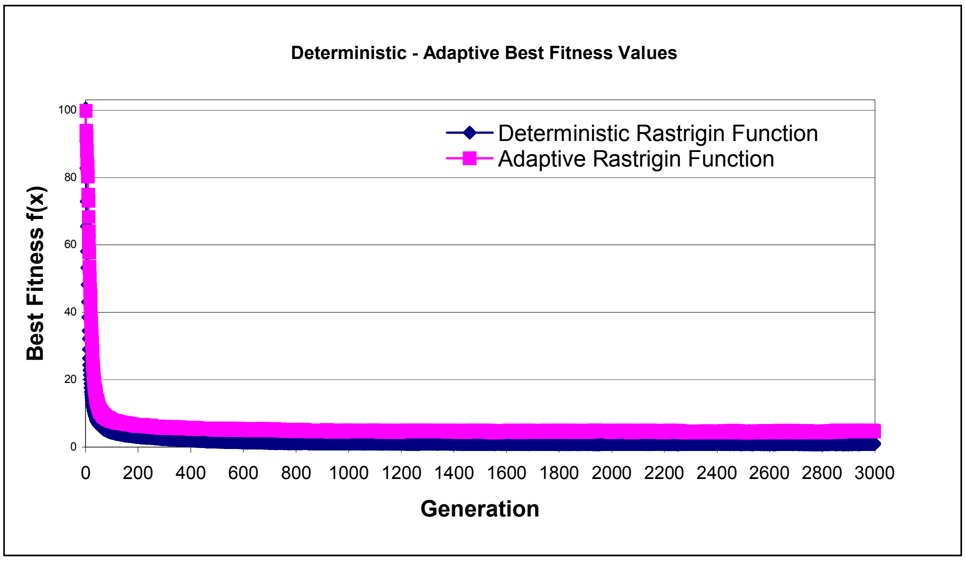

Figure 1: Deterministic vs Adaptive Best Fitness Values

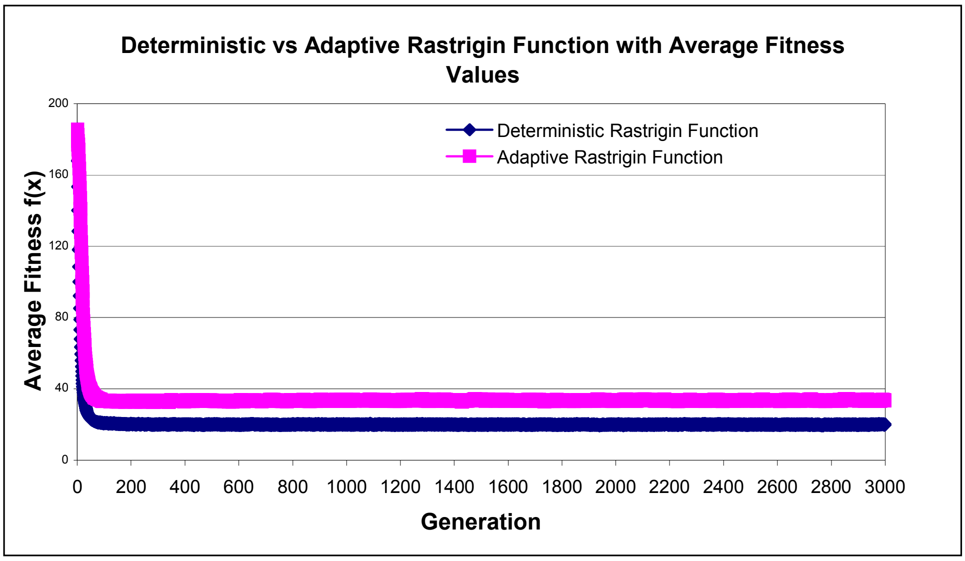

Figure 2: Deterministic vs Self-Adaptive Rastrigin Function with Average Fitness Values

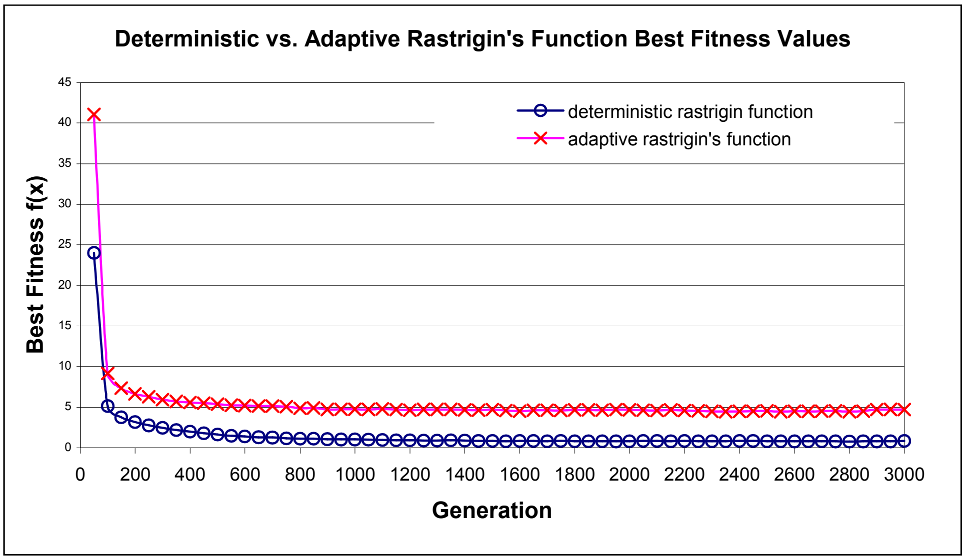

Figure 3: Deterministic vs. Adaptive Rastrigin's Function with Best Fitness Values

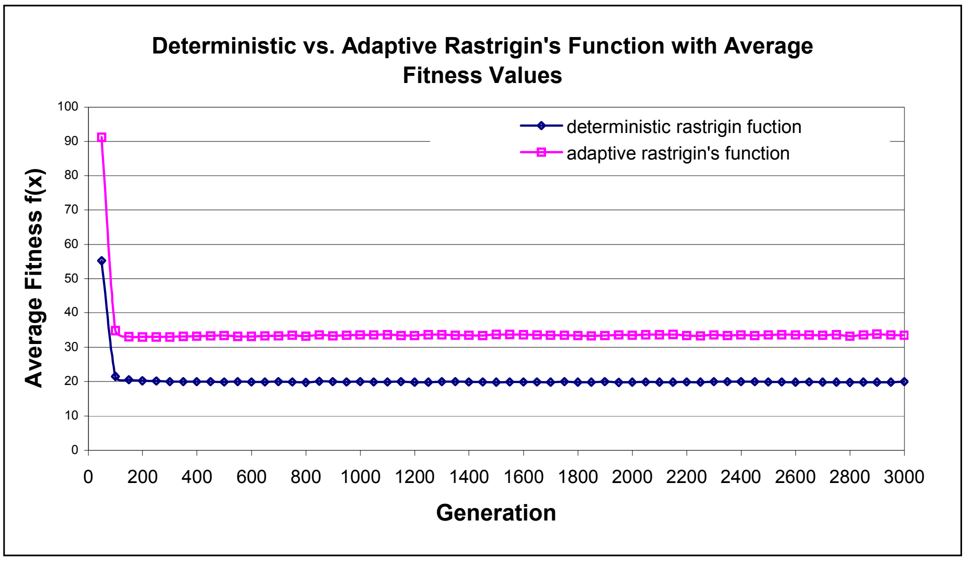

Figure 4: Deterministic vs. Adaptive Rastrigin's Function with Average Fitness Values

It is obvious that deterministic method is better than the self-adaptive method. We expect from the self-adaptive method to show better mean and best fitness values than the deterministic one. However, opposite our expectations, we observe that deterministic is better. There is no need to make a T-test for these results, because as stated above, deterministic method is better at all instances. However, we examine the mean and standard deviation of these two samples with T-tests. Mean and standard deviation is also better in the deterministic method. The same results are also observed with One-Max problem. Results are given in Table-2. We find the maximum fitness value 100 at each run both in self-adaptive and deterministic method for the One-Max Problem. We could not find the best result in Generalized Rastrigin's Function, but the best value close to zero.

	Mean of the Generation Number with Best Fitness Values
Deterministic One-Max Problem	57.221
Self-Adaptive One-Max Problem	116.21

Table 2: Statistical Values for One-Max Problem

The deterministic methods find the best values better than self-adaptive methods. Secondly, deterministic methods find the optimal values faster than self-adaptive. We see the same results from the T-tests. The standard deviation of self-adaptation is greater than the deterministic. Therefore, the values of self-adaptive method are spread through greater range than deterministic method, with greater mean and median.

For the reason given above, we observe the mean mutation rates compared to the average fitness. The results are given in Figure 5 for One-Max Problem. We see that as the mutation rate decreases, the fitness value increases. Since self-adaptive mutation begins with a higher mutation ratio, it takes more time to reach to the optima. However, there is a possibility to reach to the local optima for the deterministic algorithm because of the fast converges. Because the mutation rate is high, more genes of the bit wise chromosomes of the individuals' change, that causes it to slow down, and tracing a wide range of values (spread to the wide range), that cause greater standard deviation then deterministic method.

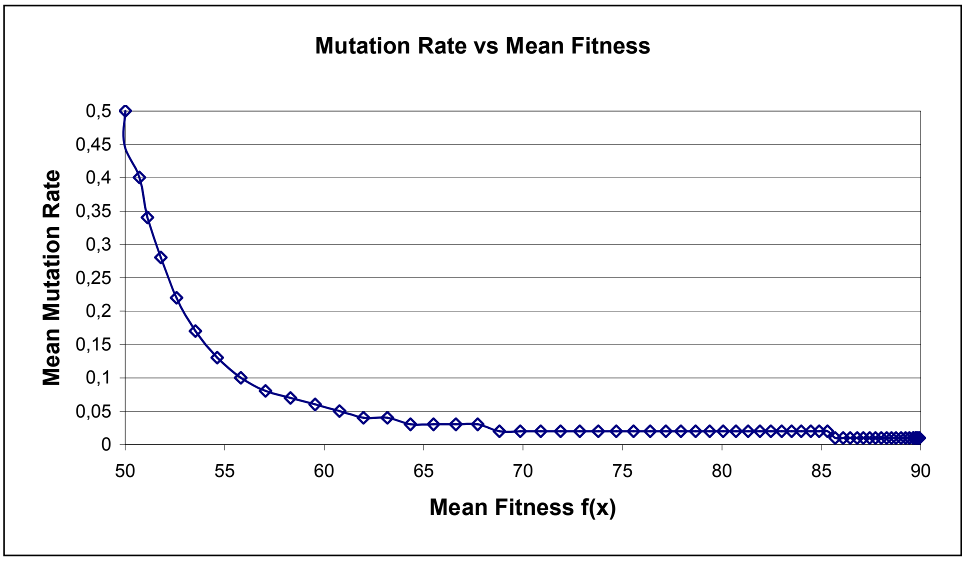

Figure 5: Mutation Rate vs Mean Fitness

In selection procedure, the individuals that have higher fitness values have more selection pressure. Therefore, as they are elected, their mutation rates are transferred to the offsprings, that causes the mutation rate to decrease. As the mutation rate decreases, individuals with low mutation rate have more pressure than others –as stated above- their less genes changes.

The lower mutation rates tend to be selected, as fitness increases is not surprising. "Back, defined the optimal mutation rate for a string of given fitness, p+, the mutation rate which maximizes the chance a fitness improvement due to mutation, and observed that this rate, which decreases as fitness increases, remains within the range of rates expressed in the population over the course of adaptation, suggesting that allowing mutation rates to self-adapt move search performance in the direction of optimality [4] ". However, for any string with more than half of its bits set to one, any mutation is more likely to change a one to a zero rather than a zero to a one, and thus the higher the mutation rate, the lower the expected fitness after mutation. This observations raises the possibility that once more than half the bits are set to one, selection always favors the lowest mutation rate possible, and it is only the rate at which mutation rates are themselves mutated that limits the rate of decrease of the population mean mutation rate.

Given the same selection pressure, the mutation rate favored by selection for an individual of a given fitness will be different in the case that it is the highest fitness individual in the population than in the case that it is the lowest. Selection will tend to favor a higher mutation rate for this individual when it is on the low end of the population fitness distribution because such a rate will probably be necessary for the individual to be able to place offspring in the higher end of the next population.

One-Max problem shows a common characteristic that the deleterious affect of mutation is usually more severe at higher fitness levels. Independent of the selection strength, selection will tend to favor individuals with lower mutation rates as fitness increases.

“Another way to understand the relationship between the intensity of selection and variable mutation rates is to observe that selection favors those individuals which maximize their individual reproductive success (RS), the number of surviving offspring they produce. A trait will then be favored by selection to the extent to which it correlates with reproductive success. The mutation rate that should then be favored is the one, which maximizes the expected number of offspring being of sufficiently high fitness. The higher the selection strength, the smaller this fraction will be, and thus the higher the fitness value required for offspring to survive. [4]”

5. CONCLUSION

In this paper, we solve two given problem by generational canonical algorithms with fixed mutation rate parameter and self-adaptive mutation rate parameter. These problems are the one-max problem and the generalized Rastrigin’s function. We compare the results that we obtained. Opposite of our expectations, self-adaptive methods show poorer performance than deterministic methods. We conclude the reason is that the mutation rate has selection pressure for the offsprings for the next generation. Self-adaptation methods start with high mutation rates with slow converges while the deterministic method converges to optima faster with fixed mutation rate.

REFERENCES

[1] T. Back and M. Shutz. Intelligent Mutation Rate Control in Canonical Genetic Algorithms. Fundamentals of Intelligent Systems, number 1079 in Lecture Notes in Artificial Intelligence, pages 158-167, Springer-Verlag, 1996.
[2] A.E. Eiben and Mark Jelasity. A Critical Note on Experimental Research Methodology in EC. In Proc. of the 2002 Congress on Evolutionary Computations, IEEE, pp 532-587.
[3] A.E. Eiben and at all. Parameter Control in Evolutionary Algorithms. IEEE Transactions on Evolutionary Computation, Vol. 3, pp. 124-141
[4] M.Glickman and K.Sycara. Evolutionary Algorithms: Exploring the Dynamics of Self-Adaptation. Genetic Programming 1998: Proceedings of the Third Annual Conference.
[5] M.Glickman and K.Sycara. Reasons for Premature Convergence of Self-Adapting Mutation Rates. Proc. of the 2000 Congress on Evolutionary Computation.