

# Fitness Costs of Mutation Rate Adaptation and its Application to Optimization of Dynamic Objective Functions

Jan T. Kim

Institut für Neuro- und Bioinformatik  
Seelandstraße 1a, 23569 Lübeck, Germany  
kim@inb.uni-luebeck.de

## Abstract

*Evolutionary algorithms can be used to solve complex optimization tasks. However, adequate parameterization is crucial for efficient optimization. Evolutionary adaptation of mutation rates provides a solution to the problem of finding suitable mutation rate settings. However, evolution of low mutation rates may lead to premature convergence.*

*In nature, mutation rate control coevolves with other functional units in a genome, and it is constrained because mutation rate control requires energy and resources. This principle can be captured by an abstract concept of fitness cost associated mutation rate adaptation, which can be generically applied in evolutionary algorithms. Application of this principle can be useful for addressing problems of premature convergence.*

*This contribution explores applications of this concept within the context of dynamic fitness landscapes. It is shown that fitness costs for mutation rate adaptation is no less advantageous in dynamic fitness landscapes than in static ones, and that interesting synergies can arise in conjunction with dynamics in multimodal fitness landscapes.*

**Keywords:** evolutionary algorithm, optimization, premature convergence, mutation rate, adaptation.

## 1. Introduction

Evolutionary algorithms (EAs), such as Genetic Algorithms [6, 11] and Evolution Strategies [9], are a versatile and powerful tool for optimization which is based on copying mechanisms of evolution in nature.

Generally, an EA operates by maintaining a population of candidate solutions, referred to as genomes, to a given optimization problem. Genomes are vectors with components which may be real-valued, e.g. in traditional Evolution Strategies, or discrete, as in Genetic Algorithms and also in nature. The population evolves for several time

steps, or generations. In each generation, the genomes are evaluated by a fitness function, which is (or simulates) the objective function of optimization. In the subsequent selection step, genomes are selected for reproduction, resulting in an offspring genome. The probability of a genome to be selected depends on its fitness value. In asexual reproduction, an offspring genome is produced by copying a selected genome. In sexual reproduction, two parent genomes are selected and an offspring genome is produced from them by using an operator which simulates natural genetic processes, such as meiosis and zygote formation, crossover or DNA recombination. Finally, the offspring genomes are subjected to mutation, i.e. they are subjected to random changes. The population obtained by applying these operations is represents the next generation.

These basic components of an EA can be embellished in many different ways. The fitness function may either be static during the evolutionary process or it may be dynamic, i.e. it may change as evolution proceeds. There exist many selection operators, such as roulette wheel or tournament selection. Typical mutation operators include addition of noise with a Gaussian distribution for real-valued genome components and bit flipping, or replacement with uniformly distributed random values, for discrete components. The operators employed in an evolutionary algorithm need to be parameterized. Finding parameter settings that ensure efficient optimization can be a problem in itself. One way to address this issue is to replace static control parameters with additional genome components, such that the suitable values for the control parameters may arise by evolutionary optimization. This concept is pursued for mutation rate settings in this contribution.

Mutations are indispensable as generators of diversity in EAs. With too little mutation, optimization may be slow and inefficient, and more importantly, the EA may be prone to premature convergence, i.e. the population may converge on some insignificant local optimum of the fitness function. However, with too much mutation, the performance of evolutionary algorithms quickly deteriorates be-

cause error thresholds prevent any significant convergence from occurring. The choice of appropriate mutation rates is, therefore, a difficult issue, and it is an attractive concept to let mutation rates coevolve along with genomes.

Adaptation of mutation rates is traditionally used in Evolution Strategies [9], and thus, it has a long history in evolutionary algorithms. A variety of schemes and concepts has been developed [1, 2, 4, 5, 8]. Such approaches always have to address the problem of premature convergence resulting from the evolution of very low mutation rates. In nature, arbitrarily low mutation rates do not arise because mutation rate control is associated with a cost in energy and resources. This observation motivates integration of a coevolutionary link between mutation rate adaptation and fitness into an evolutionary algorithm framework.

The basic principle of such a mechanism has been formalized as energy dependent, or fitness dependent mutation rate adaptation [3]. Using canonical, static fitness functions, it was demonstrated that this concept can be applied in quite a generic way. However, fitness functions are dynamic in nature. Furthermore, in all technical applications of evolutionary algorithms in which continued, on-line adaptation and optimization are required, fitness functions are inherently dynamic. Therefore, fitness dependent mutation rate adaptation is investigated in the context of dynamic fitness functions in this contribution.

## 2. Methods

### 2.1. Fitness Dependent Mutation Rate Adaptation

Fitness cost associated mutation rate adaptation (henceforth abbreviated FCMA) is implemented as described in [3]. Genomes consist of a vector  $\vec{x} = (x_0, x_1, \dots, x_{d-1})$ ,  $x_i \in \mathbb{R}$  and a vector of mutation rate modifiers  $\vec{\mu} = (\mu_0, \mu_1, \dots, \mu_{d-1})$ ,  $\mu_i \in \mathbb{R}$  where  $d \in \mathbb{N}$  is the dimension of genome space.  $\vec{x}$  is used for primary fitness evaluation (see below) while the components  $\vec{\mu}$  control mutation rates: The standard deviation of Gaussian noise added to component  $x_i$  during mutation is given by

$$m_i = m_{\text{global}} \cdot q^{\mu_i} \quad (1)$$

The global mutation rate  $m_{\text{global}}$  and the mutation modification factor  $q$  are control parameters that are set by the user. Setting  $q = 1$  effectively turns off mutation rate adaptation. The mutation rate modifiers  $\mu_i$  are mutated with a fixed, user-defined standard deviation denoted by  $M$ .

The fitness cost is calculated by rescaling the fitness values in the population to the interval  $[0, 1]$  according to

$$F_r(\vec{x}) = \frac{F_{\text{raw}}(\vec{x}) - F_{\text{min}}}{F_{\text{max}} - F_{\text{min}}} \quad (2)$$

where  $F_{\text{raw}}(\vec{x})$  denotes the fitness of the genome according to the target fitness function,  $F_{\text{max}}$  is the maximal raw fitness in the population and  $F_{\text{min}}$  is the minimal raw fitness. The effective fitness of a genome, used for selection, is then given by

$$F_{\text{eff}}(\vec{x}, \vec{\mu}) = F_r(\vec{x}) - p \cdot \sum_{i=0}^{d-1} |\mu_i| \quad (3)$$

where  $p$  denotes the mutation modification penalty, a user controlled parameter. New generations are constructed through tournament selection with tournament size 2 and asexual reproduction on the basis of  $F_{\text{eff}}$  values. The population size, i.e. the number of individuals in a generation, is user-defined and constant during a run.

Rescaling fitness values before applying the fitness penalty allows setting  $p$  without consideration of the range of values returned by the fitness function and thus facilitates comparison of results observed with different fitness functions. Setting  $p = 0$  allows mutation rate adaptation without any fitness cost.

### 2.2. Dynamic Fitness Landscapes

Dynamic fitness landscapes were derived from static landscapes by introducing a vector  $\vec{g} = (g_0, g_1, \dots, g_{d-1})$  that can be thought of as a moving center of genome space. Initially,  $\vec{g}$  is set to  $\vec{0}$ . In each time step,  $\vec{g}$  is modified by adding Gaussian noise with a standard deviation of  $D$ , where  $D$  is a user controlled parameter. The raw fitness values are calculated by

$$F_{\text{raw}} = f(\vec{x} - \vec{g}) \quad (4)$$

where  $f$  is the target fitness function. This method allows to control the speed of dynamic change in the fitness function to be controlled only by  $D$ ; if  $D = 0$ , the fitness landscape is static and dynamic change increases with  $D$ . This approach is similar to the one used in [7] in that dynamics are generated by a random walk of  $\vec{g}$ , therefore, the parameter  $D$  is referred to as the random walk speed.

For the present evaluation, the set of test functions for evolutionary algorithms already used in [3] are employed as target functions:

$$f_1 = \|\vec{x}\| \quad (5)$$

$$f_{10} = -\cos(2\pi\|\vec{x}\|) + 0.1 \cdot \|\vec{x}\| + 1 \quad (6)$$

$$f_6 = \sum_{j=0}^{d-1} (x_j^2 - 10 \cos(2\pi x_j) + 10) \quad (7)$$

$f_1$ ,  $f_{10}$  and  $f_6$  are called sphere, Salomon's and Rastrigin's test function, respectively, function indexes are adopted from [10]. For all these functions, the optimization goal

Dimension of genome space and fitness function	$d = 8$
Global mutation rate	$m_{\text{global}} = 0.1$
Global mutation rate for mutating mutation modifiers	$M = 1.0$
Mutation rate modification factor	$q = \begin{cases} 2 & \text{(FCMA active)} \\ 1 & \text{(control)} \end{cases}$
Mutation rate modification penalty	$p = 0, \dots, 1$
Mutation rate for $\vec{g}$	$D = 0, \dots, 1$
Population size	$n = \begin{cases} 500 & \text{(run series)} \\ 100 & \text{(surveys)} \end{cases}$
Number of generations	$t_{\text{max}} = 1000$

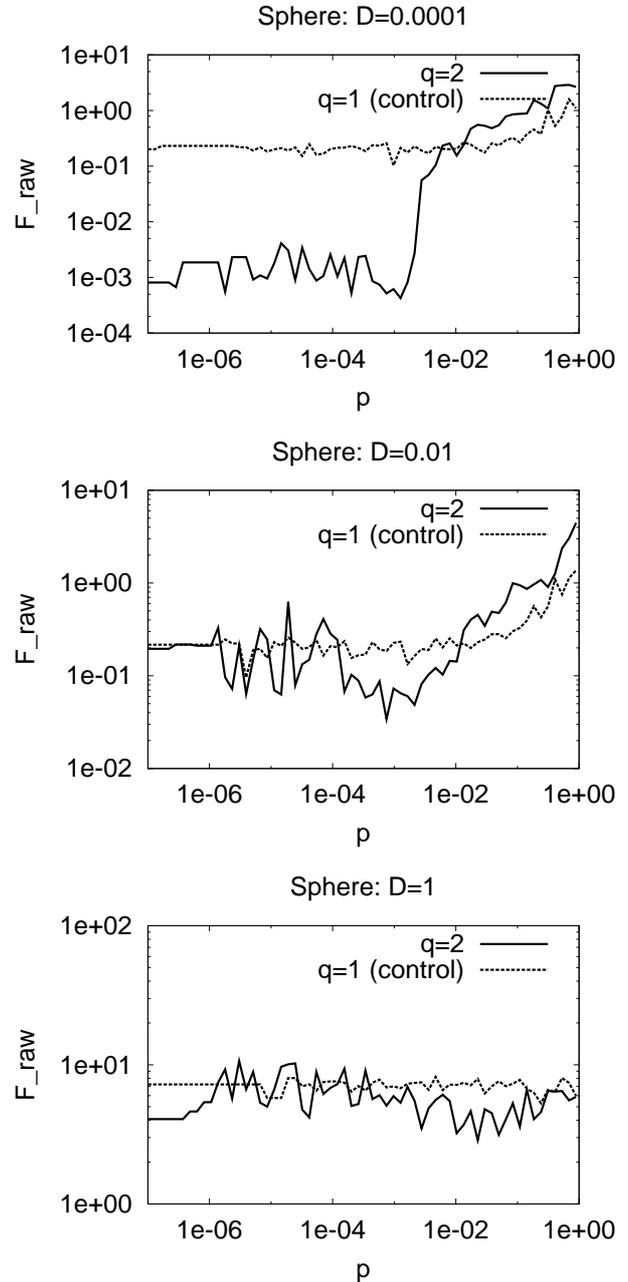
**Table 1. Control parameter settings for evolutionary algorithm runs**

is minimization, and the global optimum is located at the coordinate origin. The sphere function has no additional, local optima. Salomon’s function has sphere-shaped local optima arranged in layers around the global optimum. Rastrigin’s function has multiple local optima which form a regular, orthogonal grid.

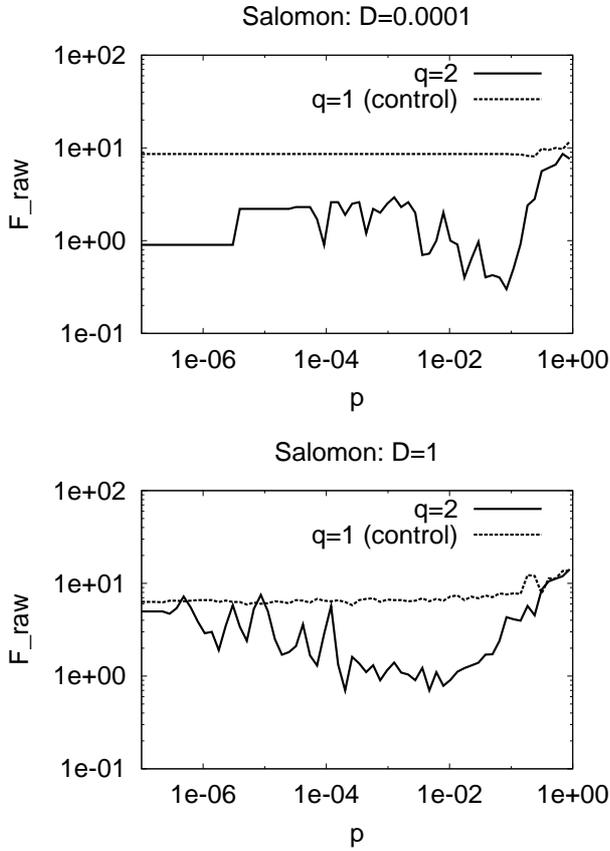
### 3. Results and Discussion

The effects of making fitness landscapes dynamic were characterized in two ways. Firstly, series of runs were performed with mutation rate modification penalties ranging from  $10^{-7}$  to 1, as in [3], and additionally, series with random walk speed settings between  $10^{-7}$  and 1 have been carried out. With values greater than 1, the steps of  $\vec{g}$  would become larger than the distance between local minima in functions  $f_{10}$  and  $f_6$ , therefore, such values were not used. Secondly, survey plots in which the plane spanned by both control parameters is scanned were performed. The control parameters are summarized in Table 1.

Figure 1 shows results obtained with increasing penalties and different random walk speeds. With slow speeds of up to  $10^{-4}$ , the characteristics are similar to those found with the static version of the sphere function: Up to a threshold value of  $p$  between  $10^{-3}$  and  $10^{-2}$ , performance is much better with mutation rate adaptation than in the control, the best results are seen with  $p$  settings less than one decimal order of magnitude below this threshold. As random walk speed increases, differences between the runs with active FCMA and the control runs disappear. The fact that higher mutation rates are more favourable when the optimum moves faster, and therefore the selective advantage of actively lowering mutation rates is less prominent, is at least partly responsible for this effect.



**Figure 1. Minimal  $F_{\text{raw}}$  values after 1000 generations with the sphere function with  $D$  set to  $10^{-4}$  (top),  $10^{-2}$  (middle) and 1 (bottom). Note that both axes have logarithmic scales.**



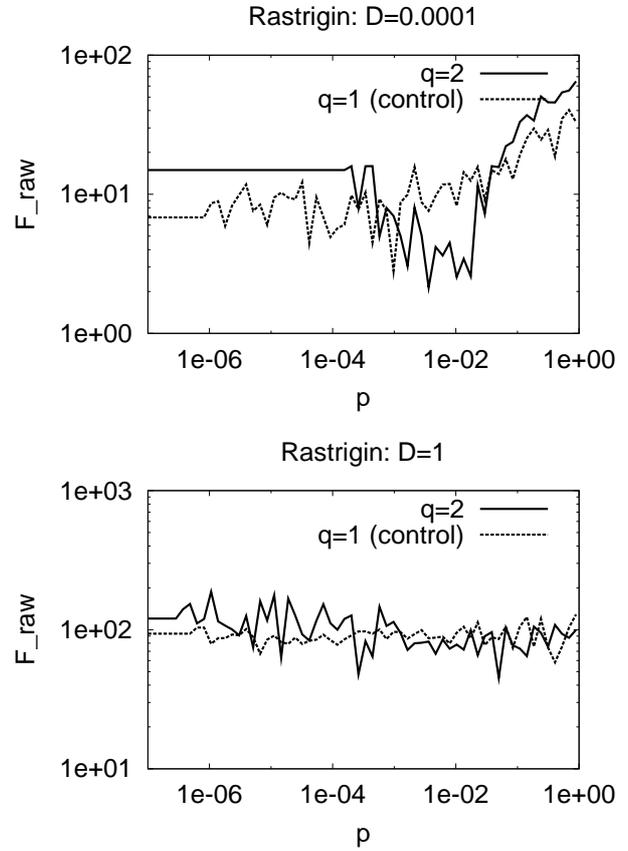
**Figure 2.** Minimal  $F_{\text{raw}}$  values after 1000 generations with Salomon's function with  $D$  set to  $10^{-4}$  (left) and 1 (right).

Contrasting to this, Fig. 2 shows that a significant difference between runs with active FCMA and controls is observed with Salomon's function even with  $D = 1$ . These differences are most prominent with intermediary penalty settings. With low as well as with high penalties, performance approaches the levels which are also obtained with no adaptation at all.

For Rastrigin's function, increases in random walk speed result in a decay of difference between runs with active FCMA and the corresponding controls, as shown in Fig. 3, similarly to the decay seen with the sphere function.

Fig. 4 shows a sweep of the random walk speed with  $p$  fixed to  $10^{-4}$ . The plot shows that with the sphere function, increasing mobility of  $\vec{g}$  results in a proportional increase in minimal  $F_{\text{raw}}$  values. This just reflects the distance by which  $\vec{g}$  moves per generation. As the sphere function has no local optima, the population tracks the minimum as closely as possible.

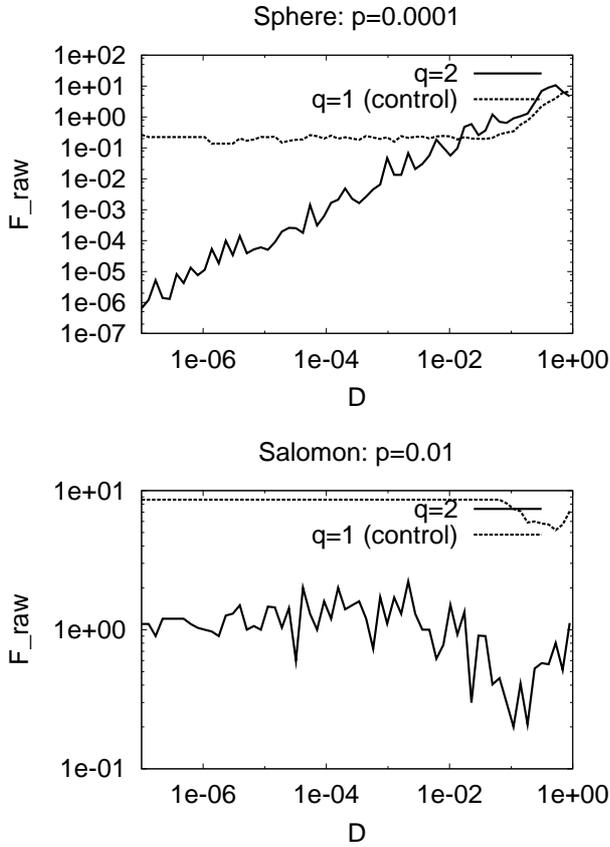
With Salomon's function, the situation is different. As



**Figure 3.** Minimal  $F_{\text{raw}}$  values after 1000 generations with Rastrigin's function with  $D$  set to  $10^{-4}$  (left) and 1 (right).

Fig. 4 reveals, mobility in  $\vec{g}$  actually leads to improvements rather than to deterioration in performance. Apparently, shifts in  $\vec{g}$  can help the population to cross maxima which are layered around the global minimum in Salomon's function. This effect is even noticeable in the controls, but the runs with active FCMA benefit more. More importantly, the range of random walk speed settings in which this favourable effect arises is much larger with active FCMA.

For Rastrigin's function, there is just a rather narrow window of penalty settings in which the runs with active FCMA perform better than the controls. Fig. 5 shows a random walk speed sweep for  $p = 10^{-2}$ , which is within this window (see Fig. 3). With  $D$  values below a threshold at approximately  $D = 10^{-2}$ , most runs with active FCMA yield better results than those without. However, this difference is much smaller in relation to the magnitude of random fluctuations than the differences seen with the other two fitness functions, as evidenced by three instances where the run with no FCMA actually turns out to



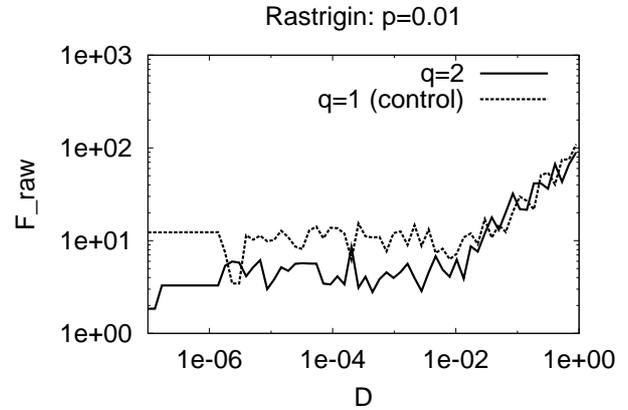
**Figure 4. Minimal  $F_{\text{raw}}$  values after 1000 generations with the sphere function with  $p = 10^{-4}$  (left) and with Salomon's function with  $p = 10^{-2}$  (right).**

fare better. These observations appear to indicate an advantage of active FCMA which is significant but much smaller than in the cases of the sphere and Salomon's function.

With  $D$  values beyond the threshold, mobility in  $\vec{g}$  results in deterioration in runs with active FCMA as well as in the controls, similar to the situation for the sphere function. This transition marks a point at which the grid pattern of local minima in Rastrigin's function becomes so blurred by the random walk of  $\vec{g}$  that the fitness landscape becomes indistinguishable from the sphere function for the population.

All these observations are summarized by the survey plots shown in Fig. 6. These analyses were performed with a population size of 100 instead of 500, after it was verified smaller populations do not lead to qualitatively different results.

For all three test functions, a trench marking penalty settings resulting in optimal minimization of  $F_{\text{raw}}$  values



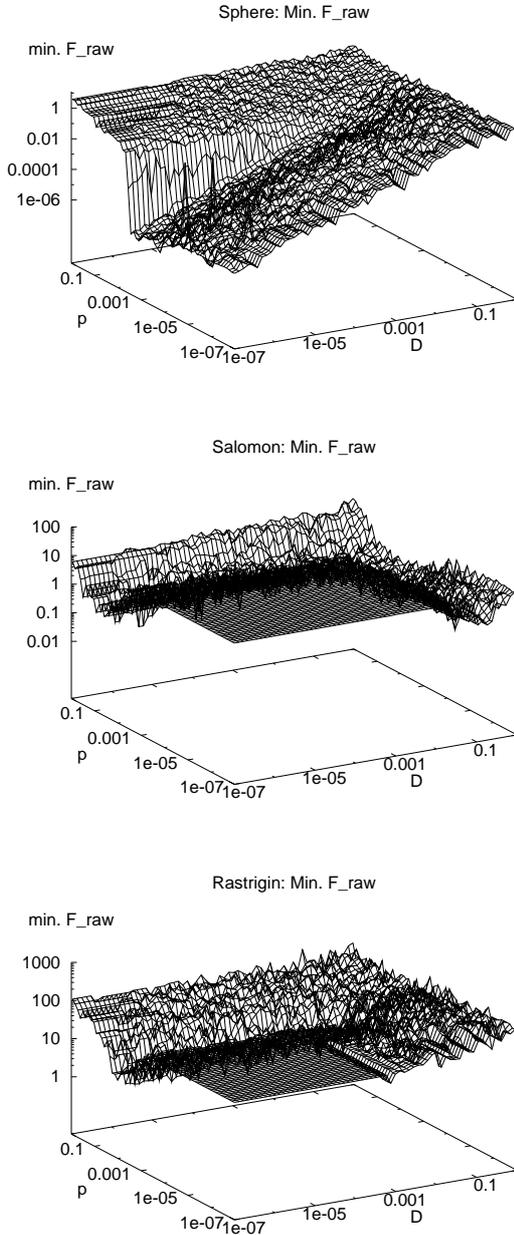
**Figure 5. Minimal  $F_{\text{raw}}$  values after 1000 generations with Rastrigin's function with  $p = 10^{-2}$ .**

can be seen at intermediate penalty settings. For penalties above this optimum, performance quickly deteriorates, most pronouncedly with the sphere function.

Increases in random walk speed have qualitatively different effects for functions without and with local minima. For the smooth sphere function, the random walk speed gives rise to a lower limit of the  $F_{\text{raw}}$  values that can be attained. Thus as speed increases, the advantage of FCMA linearly declines. This leads to a ramp-like appearance of the plot shown in Fig. 6.

With both rugged fitness functions, a plateau appears in place of a ramp. This represents a parameter range in which convergence in suboptimal minima occurs. As the plateau level is at rather large  $F_{\text{raw}}$  values, the border to the trench at intermediate penalty settings is much more pronounced than in the sphere case. The trenches in the Salomon and Rastrigin plots are also shifted towards larger penalty values compared to the sphere plot. This is related to the fact that the  $F_{\text{raw}}$  values attained with the rugged fitness functions are some orders of magnitude above those reached with the sphere function.

With Salomon's function, mobility of  $\vec{g}$  actually improves minimization, which results in formation of a second trench, orthogonal to the first one, at rather large random walk speeds. In the area where both trenches join, a widened basin of improved minimization is formed. This is an example of a favourable effect that synergistically emerges from the interplay between dynamics in the fitness landscape and FCMA. No such phenomenon is observed with Rastrigin's function, however.



**Figure 6. Minimal  $F_{\text{raw}}$  values observed after 1000 generations in populations of 100 genomes with the sphere function (top), Salomon's function (middle) and Rastrigin's function (bottom). Values are plotted as a function of  $p$ , the mutation rate modification penalty, and  $D$ , the mutation rate for  $\vec{g}$ . Note that all three axes have logarithmic scales.**

## 4. Conclusions

Using a simple and straightforward method for enhancing test fitness functions with temporal dynamics, it was shown that FCMA can successfully be used with dynamic fitness landscapes. With the unimodal sphere fitness function, the effects of introducing a dynamic element are quite trivial, while more complex phenomena are observed in conjunction with multimodal fitness functions.

FCMA should be considered as a novel method for controlling search. With respect to the traditional Evolution Strategy approach, the fitness cost introduces a coevolutionary link between the genome part  $\vec{x}$  which represents the candidate solution to the objective function, and  $\vec{\mu}$ , the part of the genome which controls mutation rates. This link is conveniently parameterized with  $p$ . The results presented here confirm that associating mutation rate adaptation with a fitness cost can prevent premature convergence.

FCMA is similar to the approach described in [5], where a globally prescribed, fixed formula is used to down-regulate mutation rates for genomes that are already well adapted. In comparison to this approach, FCMA is more flexible, as it permits individual allocation of the raw fitness resource between mutation rate control and competition with others. Coevolving mutation rates and applying a fitness cost is also closer to the natural evolution process than externally assigning differential mutation rates.

Despite being gleaned from nature, evolution with FCMA is subject to limitations, as all other methods are. This is illustrated by the results seen with the sphere function: Of course, the precision limit for minimization induced by the random walk of  $\vec{g}$  cannot be overcome.

However, such a strict and monocausal limitation is a rather special case. This is illustrated by the other fitness functions, which have structural features that give rise to limitations which may dominate those induced by dynamics. The combined effects of FCMA and dynamics in the fitness landscape may be favourable, as seen in the case of Salomon's function. The results obtained with Rastrigin's function, on the other hand, show that such favourable conditions cannot be generally expected.

## 5. Future Work

As a next step, it is planned to test FCMA with fitness landscape variants in which  $\vec{g}$  moves directionally instead of performing a random walk. It would also be interesting to explore other classes of dynamic fitness landscapes, such as oscillating landscapes [12]. It would be desirable to develop a theoretical basis for understanding the possible synergy between FCMA and fitness landscape dynamics.

The study presented here was restricted to asexual reproduction to ensure that mutation is the only source of

new genotypes, and thus of diversity. This allows investigating the effects of FCMA in isolation, without interference with other sources of diversity, such as sexual reproduction. However, sexual reproduction is pivotal in many applications of evolutionary algorithms, and therefore, combination of FCMA with operators for sexual reproduction should be investigated in the future.

Finally, it would be desirable to systematically characterize the changes in the evolutionary process caused by introducing the fitness cost for mutation rate adaptation. This might be helpful to better assess for which optimization purposes the fitness cost approach is a suitable tool. Additionally, such an analysis may improve our understanding of the conditions that shape evolution in nature.

## 6. References

- [1] T. Bäck. The interaction of mutation rate, selection, and self-adaptation within a genetic algorithm. In R. Männer and B. Manderick, editors, *Parallel Problem Solving from Nature*, 2, pages 85–94, Amsterdam, 1992. Elsevier Science Publishers.
- [2] T. Bäck. Optimal mutation rates in genetic search. In S. Forrest, editor, *Proceedings of the Fifth International Conference on Genetic Algorithms*, pages 2–8, San Mateo, CA, 1993. Morgan Kaufmann Publishers.
- [3] J. T. Kim. Energy dependent adaptation of mutation rates in computer models of evolution. In C. Adami, R. K. Belew, H. Kitano, and C. Taylor, editors, *Artificial Life VI*, pages 248–255, Cambridge, MA, 1998. MIT Press.
- [4] C. Maley. The coevolution of mutation rates. In F. Morán, A. Moreno, J. J. Merelo, and P. Chacón, editors, *Advances in Artificial Life*, pages 219–233, Berlin Heidelberg, 1995. Springer Verlag.
- [5] S. Marsili Libelli and P. Alba. Adaptive mutation in genetic algorithms. *Soft Computing*, 4:76–80, 2000.
- [6] M. Mitchell. *An Introduction to Genetic Algorithms*. MIT Press, 1996.
- [7] M. Nilsson and N. Snoad. Error thresholds on dynamic fitness-landscapes. working paper 99-04-030, Santa Fe Institute, 1399 Hyde Park Road, Santa Fe, New Mexico 87501 USA, 1999.
- [8] A. Ostermeier. An evolution strategy with momentum adaptation of the random number distribution. In R. Männer and B. Manderick, editors, *Parallel Problem Solving from Nature*, 2, pages 197–206, Amsterdam, 1992. Elsevier Science Publishers.
- [9] I. Rechenberg. *Evolutionsstrategie '94*. Frommann-Holzboog, Stuttgart-Bad Cannstadt, 1994.
- [10] R. Salomon. Re-evaluating genetic algorithm performance under coordinate rotation of benchmark functions. a survey of some theoretical and practical aspects of genetic algorithms. *BioSystems*, 39:263–278, 1996.
- [11] M. D. Vose. *The Simple Genetic Algorithm*. MIT Press, Cambridge, MA, 1999.
- [12] C. O. Wilke. *Evolutionary Dynamics in Time-Dependent Environments*. PhD thesis, Institut für Neuroinformatik, Ruhr-Universität Bochum, 1999. ISBN: 3-8265-6199-6.