# Strategy Parameter Variety in Self-Adaptation of Mutation Rates

**Christopher Stone**
Intelligent Computer Systems Centre
University of the West of England
Bristol, United Kingdom

christopher.stone@uwe.ac.uk

**Jim Smith**
Intelligent Computer Systems Centre
University of the West of England
Bristol, United Kingdom

james.smith@uwe.ac.uk

## Abstract

Self-adaptation has been widely used in Evolution Strategies (ES) and Evolutionary Programming (EP), where it has proved useful in varying the mutation step size for continuous objective variables. To date, relatively little work has been performed on applying self-adaptation to the canonical Genetic Algorithm (GA). This research applies a simple discrete model of self-adaptation to test functions with differing characteristics. We show that the discrete model is able to provide more reliable problem solving than the classical lognormal self-adaptation scheme on the test problems examined. We find that although self-adaptation parameter choices representing conventional thinking perform best for unimodal functions, very different parameter settings are required for optimum performance on multimodal functions. These results are discussed in terms of the strategy parameter variety needed for self-adaptation to work effectively and we outline a self-adaptation mechanism designed to capitalize on these findings.

## 1 INTRODUCTION

In a self-adaptive Evolutionary Algorithm (EA), the representation for individuals in the population is extended to include strategy parameter information. The EA operates as normal, evolving the population according to the fitness of its members, with the additional step of stochastically varying the strategy parameters of individuals selected for reproduction. Self-adaptation of mutation rates is possibly the most common application of self-adaptation, largely stemming from its widespread use in ES (Schwefel, 1981) and EP (Fogel, Fogel & Atmar, 1991). For the purposes of self-adaptation, the main difference between GAs and ES/EP is that GAs usually employ a binary representation. With such a representation, a per-bit mutation rate is used to control the rate of bit-flipping mutations applied to an individual. For a non-adaptive GA, this parameter is fixed across the population and throughout the course of a run. However it is natural extension to encode the mutation rate into each individual, to allow it to vary across the population and in time. Bäck (1992) used these ideas and performed seminal work showing that self-adaptation in GAs is possible. Following Bäck's work, several authors have experimented with self-adaptation of mutation rates in GAs (see for example, Bäck & Schütz, 1996; Smith & Fogarty, 1996; Hinterding, 1997). Design decisions that must be addressed with this approach are the choice of representation for the strategy parameter and, related to this, the means by which the strategy parameter is itself varied to allow adaptation to occur. Bäck's early work remained close to the traditional interpretation of a GA and used a binary encoding of the strategy parameters with corresponding bitwise mutation. Current thinking is that a real-valued representation is preferable (Glickman & Sycara, 1998). This then allows the use of lognormal adaptation of strategy parameters as shown in (1) where the $\tau$ parameter controls the step size of $\sigma_i$, the individual's mutation rate.

$$\sigma_i' = \sigma_i \cdot \exp(\tau \cdot N(0,1)) \qquad (1)$$

Recent empirical (Liang et al. 1998; Glickman & Sycara 2000) and theoretical (Rudolph 1999) work has shown that self-adaptation schemes which adapt too quickly can lead to premature convergence to low step sizes, with the search getting 'stuck' at local optimum. This has lead to an interest in alternative variation schemes.

Smith (2001) introduces a dynamical systems model of a GA with self-adaptation of mutation rates. The model is

Table 1: Evaluation Functions

| | Name | Formula | Length (bits)/ Encoding | Optimum | Allowed Generations | |
|---|---|---|---|---|---|---|
| | | | | | w/o xover | w/xover |
| f1 | OneMax | $\sum_{i=1}^{l} x_i$ | 128 | 128 | 100 | 500 |
| f2 | Inverted Rastrigin's Function | $\frac{1}{\sum_{i=1}^{l/16}[x_i^2 - 10\cos(2\pi x_i) + 10] + 1}$ <br> $-5.12 < x_i < 5.12$ | 64 <br> 4 dimensions each 16 bits | 1 | 500 | 500 |
| f3 | Deb's Fully Deceptive Function | $\sum_{j=1}^{l/4} \begin{cases} 0.2\cdot(3-b_j), b_j<4 \\ 1, b_j=4 \end{cases}$ where $b_j = \sum_{i=1}^{4} x_i$ | 32 | 8 | 250 | 100 |
| f4 | Matching Bits | $\sum_{i=1}^{l-1} \begin{cases} 1, x_{i+1}=x_i \\ 0, x_{i+1}\neq x_i \end{cases}$ | 24 | 23 | 1000 | 100 |
| f5 | Royal Road R1 | $\sum_{j=1}^{l/8} \begin{cases} 0, b_j<8 \\ 8, b_j=8 \end{cases}$ where $b_j = \sum_{i=1}^{8} x_i$ | 64 | 64 | 1000 | 1000 |

used to predict mean fitness of an evolving population over time. To make the mathematics computationally tractable, there are two key differences between the model and the self-adaptive GAs just described. Firstly, rather than using a binary or real-valued representation, strategy parameters are represented by a single allele of alphabet $q$, where $q$ is small. Smith uses a value of 10 and this is also the value used in the present work. A consequence of this is that the mutation rate attached to an individual can only take on one of $q$ possible values, as opposed to the large or effectively infinite number available with binary or real-valued representations. Secondly, because of the discrete nature of the strategy parameter representation, the lognormal scheme in (1) cannot be used to vary the strategy parameters. Although it would be possible to provide some discrete variant of this algorithm, for simplicity Smith uses a scheme that modifies every individual's strategy parameter with probability $z$ and equal likelihood of changing to each of the $q$ possible alleles. Because it is possible for the modified strategy parameter to retain the same allele as the original one, the probability $P_a$ that the strategy parameter is altered is given by:

$$P_a = z \cdot (q-1)/q \qquad (2)$$

Control over the degree of change of the inherited strategy parameter is provided by $z$, an external parameter known as the *innovation rate*. This provides the variation needed for selection to choose preferable strategy parameters. The conventional view is that a non-uniform distribution is desirable, for example, the lognormal scheme in (1), to generate many small perturbations of strategy parameter value with larger variations being possible, but less likely. This is in order to provide occasional large changes in value to prevent the EA from getting stuck when searching rugged landscapes. As presented, Smith's model represents a considerable departure from this view since it uses a uniform distribution that provides equal opportunity for large or small perturbations[1]. In the present work we examine the implications of this variation scheme and the effects of varying the innovation rate on the performance of the GA. We do not attempt to demonstrate the need for self-adaptation, as this has been done elsewhere (see for example, Stephens et al, 1998).

## 2 EXPERIMENTAL SETUP

The GA used for the experiments provides a real-valued strategy parameter linked to each individual in the population. For discrete adaptation schemes, the strategy parameter encodes one of ten representative mutation rates in the range *minrate* to *maxrate* for the individual. Each strategy parameter is initialized to a random allele and varied according to (2). For continuous adaptation schemes, the strategy parameter encodes an arbitrary

[1] Although it should be noted that this is not a requirement of the model.

mutation rate for the individual, initialized to a random value in the range *minrate* to *maxrate* and variation of the strategy parameter is performed by the multiplicative lognormal function (1). To provide consistency with the discrete scheme, the resulting mutation rate is limited to the range *minrate* to *maxrate*.

The GA is generational because short-term regression is desirable with self-adaptation to allow adequate learning of strategy parameters (Schwefel, 1997). Selection is performed using either truncation (extinctive) selection or fitness proportionate (preservative) selection sampled using Baker's (1987) SUS algorithm. The differences between extinctive and preservative selection mechanism are discussed in (Bäck & Hoffmeister, 1991). A population of 500 individuals is maintained to reduce the variance of results and to allow adequate sampling of strategy parameters in the mating pool even with high selection pressure. (100,500) truncation selection is used, based on results from ES (Schwefel, 1981). Genetic operators are single point crossover applied with a rate of 0.7 or zero and bitwise mutation applied with the probability given by the individual's strategy parameter. All results presented are the mean of 50 runs unless otherwise noted.

Five functions with a broad range of characteristics were used for these experiments. These are detailed in Table 1.

Experiments were run on each of the functions with innovation rates of 0.01, 0.05 and 0.1 to 1 in steps of 0.1 using two selection pressures: (100,500) truncation selection and fitness proportionate selection. The ten standard mutation rates used for these experiments were 0.0005, 0.001, 0.0025, 0.005, 0.0075, 0.01, 0.025, 0.05, 0.075 and 0.1.

For each set of experiments, two metrics were used to compare results. *Reliability* was measured using the number of times the global optimum was found out of 50 runs. *Time to optimum* was used to measure the ability of the GA to solve the problem. Our stance is that a self-adaptive GA must achieve reliability before time to optimum issues can be addressed. This is particularly important in online applications of self-adaptation, where the luxury of multiple runs is not feasible, for example in process control or robotics applications.

## 3 GA WITH NO RECOMBINATION

Figure 1 shows the time to optimum of the GA for each function using a high selection pressure. The graph is annotated with reliability data, where the optimum was not found in all 50 runs. It shows that high innovation rates provide the most reliable problem solving. These results can be separated according to whether the function is unimodal or multimodal. For the purposes of this work, we classify f5 (Royal Road) as multimodal, since its fitness landscape consists of a series of peaks connected by ridges[2], which produce similar effects to the local optima of multimodal functions. Under this classification the only unimodal function in the test suite, f1 (OneMax) achieves the best performance with an innovation rate of 0.05. Higher innovation rates tend to impact performance, although not excessively so. In contrast, the multimodal functions show a trend towards both faster and more reliable performance as the innovation rate increases.

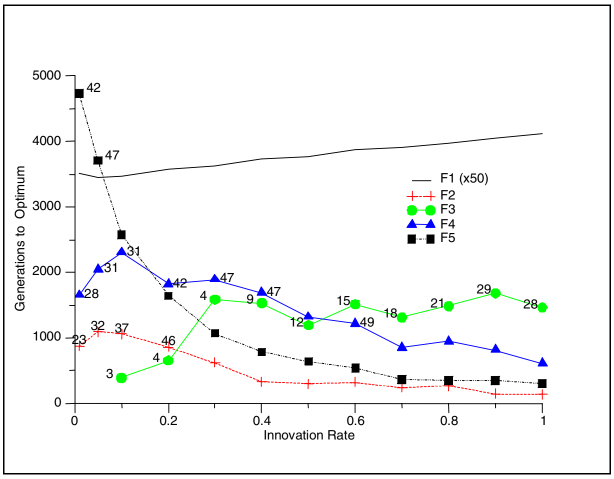


Figure 1 - Generations to Optimum against Innovation Rate for High Selection Pressure

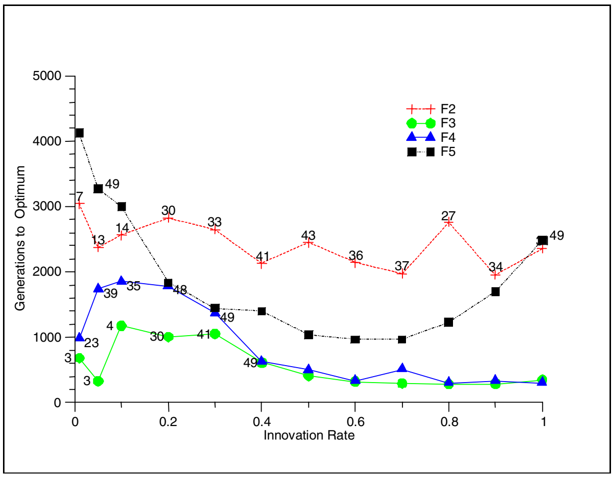


Figure 2 - Generations to Optimum against Innovation Rate for Low Selection Pressure

Results for the same set of experiments performed under low selection pressure are show in Figure 2. Results are

[2] The exact details of the fitness landscape depend on the operators employed.

not shown for f1 (OneMax) since fitness proportionate selection failed to find the optimum in any of the 50 runs. However, the low selection pressure is an advantage for f3 (Deb's Deceptive function) and reliability is much improved over high selection pressures. The graphs are somewhat noisier than their high selection pressure counterparts due to the lower number of runs in which the optimum was found for some functions and innovation rates. In general, reliability is still improved with high innovation rates, with the exception of f2 (Rastrigin's function), but unlike the case with high selection pressure, time to optimum appears to peak at an innovation rate lower than one.

## 4 GA WITH RECOMBINATION

Experiments on a GA with recombination were performed using an inheritance mechanism, which selects one of the two parental strategy parameters at random with equal probability for the offspring prior to innovation. This choice is discussed further in (Stone, 2001).

Figures 3 and 4 show the effects of innovation on reliability and time to optimum of the GA with crossover. With fitness proportionate selection the optimum for function f1 (OneMax) was never located in any of the 50 runs, so no results are shown for these experiments. With high selection pressure, reliability tends to improve as innovation rate increases whilst time to optimum remains relatively flat due to the effects of crossover. With low selection pressure, it seems that high innovation rates although still providing reliable operation are generally detrimental to time to optimum.

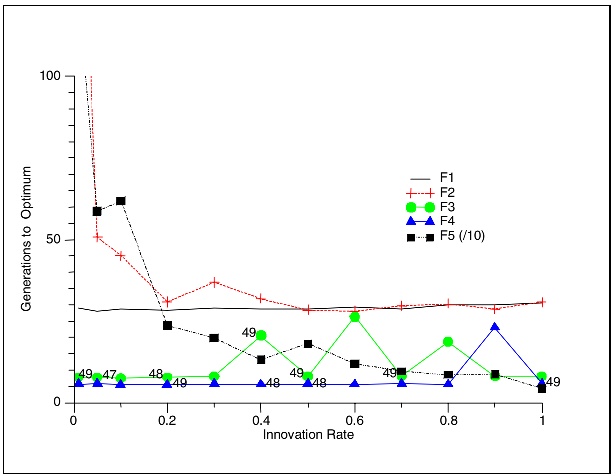


Figure 3 - Generations to Optimum against Innovation Rate for Random Inheritance with High Selection Pressure

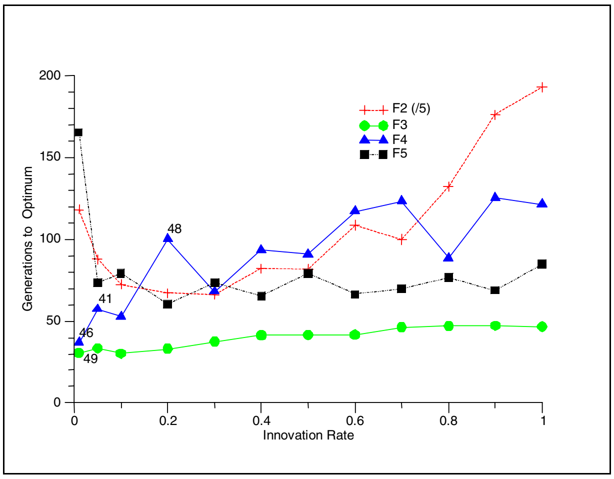


Figure 4 - Generations to Optimum against Innovation Rate for Random Inheritance with Low Selection Pressure

## 5 COMPARISON OF DISCRETE INNOVATION TO CONTINUOUS INNOVATION

To compare the performance of the discrete innovation scheme with the more mainstream continuous innovation scheme we used various settings for the continuous innovation scheme's $\tau$ parameter. The ES literature suggests the following formula for setting $\tau$:

$$\tau = c / \sqrt{n} \qquad (3)$$

where $n$ is the number of objective variables and $c$ is a problem-specific constant. However, this rule of thumb has not, to our knowledge, been exhaustively tested in a GA environment. In ES, $n$ is the number of objective variables in the representation, whereas in the case of a GA it is the length of the string in bits. It is not apparent that the rule can be directly mapped from ES to GA representations. Glickman & Sycara (1998) use a value of 0.1 for $\tau$ on a string of length 1000. This corresponds to a value of $c$ of 3.16. In contrast, Hinterding, Michalewicz & Peachey (1996) use a fixed value of 0.013 in their self-adaptive GA. In the absence of other information, we tried values for $c$ of 0.5, 1, 2, 3 and fixed rates of 0.013 and 0.02. For the former values the problem length (in bits) is taken into account when arriving at the actual rates used, whereas the latter two are fixed rates across all functions.

Table 2: Comparison of Best Results for Discrete and Continuous Self-Adaptation

| | Discrete Self-Adaptation | | | Continuous Self-Adaptation | | | |
|---|---|---|---|---|---|---|---|
| | Time to Optimum | Successful Runs | *z* | Time to Optimum | Successful Runs | $\tau$ | *c* |
| f1 (OneMax) | 68 | 50 | 0.05 | 66 | 50 | 0.063[1] | 2.00 |
| f2 (Rastrigin's) | 137 | 50 | 1 | 84 | 34 | 0.013 | 0.10 |
| f3 (Deb's Deceptive) | 1680 | 29 | 0.9 | 20 | 8 | 0.020 | 0.11 |
| f4 (Matching Bits) | 608 | 50 | 1 | 231 | 40 | 0.020 | 0.10 |
| f5 (Royal Road) | 298 | 50 | 1 | 159 | 50 | 0.013 | 0.10 |

We ran the same set of mutation-only experiments as performed for the discrete scheme, using (100,500) truncation selection and limiting each run to the same number of generations as before (see Table 1). Table 2 compares the best result for each function from the discrete and continuous schemes. Here, best is defined as the result showing most reliability followed by smallest time to optimum for the function. The discrete scheme provides reliable results and acceptable time to optimum for all functions, whereas the continuous scheme, although apparently capable of providing superior time to optimum, displays poorer reliability. It is also interesting that the best results for all of the multimodal problems arise with a value of *c* of approximately 0.1. This suggests that the relationship in (3) may also hold for GAs with a bitstring representation. However, the range of problem lengths, *l*, tested is quite restricted. Further work is needed to determine if this pattern extends to other multimodal problems and values of *l*.

# 6 DISCUSSION

## 6.1 INTERPRETATION OF RESULTS

Results for the discrete model suggest that with a high selection pressure, a low innovation rate is appropriate for unimodal problems, whereas an innovation rate of one gives the best results for multimodal problems. The former represents conventional thinking whereas the latter result is novel and requires some explanation. For low selection pressures it appears that the optimal innovation rate to achieve reliability is lower and the best performance is worse than for high selection pressure. We conducted ANOVA analysis which showed that the effects of innovation rates were a statistically significant factor in the time to optimum, and post-hoc tests with a variety of measures confirmed the difference in performance between high and low innovation rates.

In the following discussions, we assume a causal relationship between mutation rate and any resulting mutation. That is, that the change in fitness of an individual resulting from a mutation is, in general, directly related to the mutation rate in force for that individual. The innovation scheme presently used mutates strategy parameters with a uniform distribution, so any resulting value is equally likely. However, when the innovation rate is less than one, there is an increased probability for the inherited strategy parameter to escape innovation and be transmitted to the next generation. This results in a non-uniform overall distribution for the innovation mechanism, with existing mutation rates being preferred. Thus, we may classify a low innovation rate as one that exploits existing strategy parameter information. In contrast, an innovation rate of one generates each strategy parameter allele with equal probability and is explorative.

We can therefore summarize these results by saying that unimodal problems require an exploitative algorithm, whilst multimodal problems perform best with an explorative algorithm. This finding supports that of Bäck (1992) who reached a similar conclusion regarding the results of using single versus multiple strategy parameters in a self-adaptive GA.

The continuous innovation scheme does not appear to provide the same degree of reliability as the discrete scheme in mutation-only experiments. This suggests that for difficult problems, the search gets stuck in local optima, from which it cannot escape in a reasonable amount of time. The lognormal scheme perturbs the inherited strategy parameter such that the perturbation is small with high probability. Unlike the discrete scheme, which has only a few possible mutation rates, the continuous scheme provides effectively an infinite choice of mutation rates. However since the strategies are encoding for bitwise mutation probabilities, rather than step sizes, many of these rates will have very similar effects in terms of the number of allele values changed in the representation. Thus although this scheme provides variety within the population that does not exist in the discrete scheme, when we consider the likely number of bits mutated, a specific point in the search can vary its associated search strategy by a large amount with only low probability. This is comparable to the case of the discrete scheme with a low innovation rate and produces similar results.

### 6.2 PREMATURE CONVERGENCE

Multimodal problems have landscapes containing local optima and search information built up in previous generations may not be particularly useful, since the search can be attracted towards false optima. What is good for an individual at a certain stage of the search (i.e., a low mutation rate) may not necessarily be optimal for the overall search longer term, especially since the self-adaptive algorithm is inherently a greedy adaptive process, as evidenced by the need for non-elitist schemes. To counter this tendency, higher mutation rates must be available if the search is to escape from local optima.

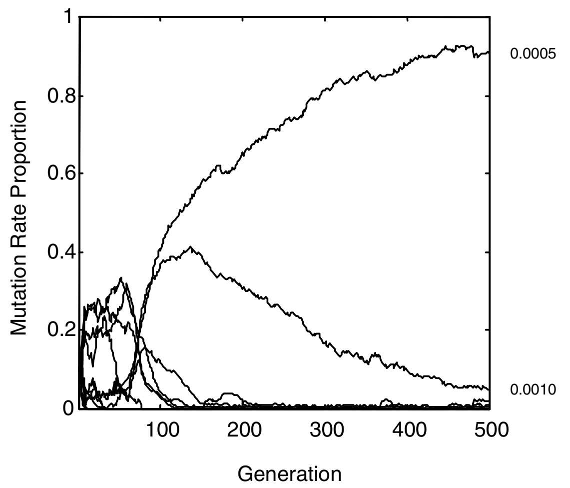


Figure 5 – Premature Convergence of Mutation Rates for f2 (Rastrigin's function)

For high selection pressure, graphs of the proportion of each strategy parameter allele present in the population with low innovation rates show that the population quickly assumes a small number of low mutation rates, typically the two or three lowest available rates. As an example, Figure 5 shows this for f2 (Rastrigin's function). Even though they are being introduced at a fixed rate by innovation, higher mutation rates exist in the population with very low, possibly zero, probability.

Liang et al (1998) and Glickman & Sycara (2000) observe premature convergence of strategy parameters with continuous self-adaptation schemes. Rudolph (1999) shows that convergence of the search to local optima can occur if the step size is reduced too rapidly[3]. At the level of the population, this appears to be the cause of poor reliability with the discrete scheme when using a low innovation rate. The fact that we see premature convergence of strategy parameters together with poor problem solving reliability using the discrete self-adaptation scheme suggests that this may be an effect common to all types of self-adaptation.

The operation of self-adaptation depends on variety of both the individual and its associated strategy parameter. Without adequate variety, self-adaptation will proceed only slowly. Variety is provided by the population and the self-adaptation algorithm to varying degrees, depending on the selection pressure and the nature of the self-adaptation scheme in use. High selection pressure has the characteristic of creating multiple copies of an individual, the strategy parameters of which are varied by innovation. With a low innovation rate and a GA operating without recombination, there are many copies of the individual produced with identical strategy parameters. Low innovation rates used with a high selection pressure are thus wasteful of function evaluations and do not represent a good approach. However, they may be a viable approach for low selection pressures, because fewer copies are produced of each individual and therefore relatively few function evaluations are wasted, especially as the nature of the low selection pressure is to preserve more of the population's variety. This view is supported by inspection of the best/mean/worst fitness (not shown) of low and high innovation rate experiments showing that mean fitness is roughly the same for both low and high innovation rates. However, the best individuals in the population are much more fit and the least fit individuals are much poorer with a high innovation rate. In short, the population shows a higher fitness variance with a high innovation rate.

### 6.3 METHODS FOR PROVIDING VARIETY

One approach is to assign a variety of strategy parameters to copies of the individual, for example by using an innovation rate of one. If the individuals are mutated and the results evaluated, now the relative fitness of the individual is determined solely by the strategy parameter (ignoring the stochastic effects of mutation). Selection is therefore evaluating the match between the individual and its assigned strategy parameter. This provides an emphasis on the appropriateness of the strategy parameter and hitchhiking of strategy parameters is discouraged. A somewhat similar approach is used in Improved Fast Evolutionary Programming (Yao, Lin & Liu, 1997). This variant of EP selects the best of two offspring generated from the same individual, one based on a step size generated from a Gaussian distribution, the other from a step size generated by a Cauchy distribution. Rudolph (1999) suggests the addition of a fixed step size in order to escape local optima. The discrete scheme with an

[3] We note that Rudolph's proof is based on an elitist EA and Rechenberg's 1/5 success rule.

innovation rate of one takes this one stage further and generates multiple offspring from the same individual with stochastically selected step sizes.

A compromise between retaining the inherited strategy parameter for testing, yet providing variety, is to pass through one copy of the current strategy parameter, together with several different choices of strategy parameter for selection to test. This scheme simultaneously allows exploration of new strategy parameters and exploitation of existing information. Based on the results of the experiments, these would seem to be desirable characteristics of any self-adaptation algorithm. Although this scheme cannot be achieved deterministically using the current discrete innovation algorithm, it is possible to calculate from (2) the probability, $P_v$, that exactly one of the $n$ (in the present case, five) copies retains the inherited strategy parameter, with the other four having different random alleles:

$$P_v = n \cdot (1 - z + z/q)(z - z/q)^{(n-1)} \qquad (4)$$

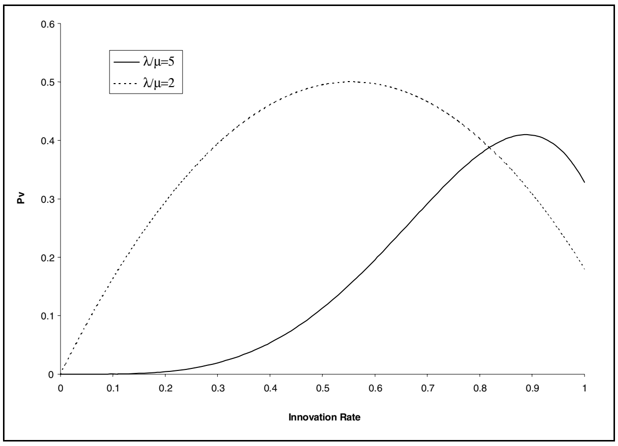


Figure 6 – Probability $P_v$ against Innovation Rate

Figure 6, a plot of this probability against innovation rate, shows that for high selection pressure, $P_v$ is negligible for low innovation rates, but increases with innovation rate, peaking at an innovation rate of approximately 0.89. An innovation rate of one, as used in the experiments gives a $P_v$ value higher than any innovation rate below about 0.74.

This demonstrates that the discrete scheme, even with an innovation rate of one, provides good sampling of innovative strategy parameters *and* still allows transmission of inherited strategy parameter information with a respectable probability. In addition, we find that for low selection pressure, when fewer copies are made of each individual, the probability distribution is substantially different, peaking at a lower innovation rate than that of high selection pressure. This is again consistent with the results reported above. The probability of the inherited strategy parameter being passed through is also much higher than with the high selection pressure over a wide range of innovation rates, such that $P_v$ is higher for low selection pressure than for high selection pressure for all innovation rates except those approaching one. This provides further reinforcement for the importance of selection pressure in the behaviour of self-adaptation.

As mentioned earlier, a deterministic version of this self-adaptation algorithm would provide for $P_v$=1 by passing through a single copy of the inherited strategy parameter, whilst testing other choices of strategy parameter. The selection pressure, $\lambda/\mu$, effectively determines the number of copies of each individual made and in the present experiments this ratio is five. However, there are ten possible mutation rates that could be sampled, meaning that some rates will not be tested against each individual in a diverse population. Clearly, it would be possible to match the number of possible mutation rates to the selection pressure in use, especially for high selection pressures. This has the additional advantage that no inheritance mechanism is needed, since all mutation rates are tested against each individual. Further work is needed to experiment with such a scheme and determine the minimum number of strategy parameter alleles that may be successfully used.

## 7 CONCLUSIONS

We showed that Smith's discrete model is able to provide effective self-adaptation in a GA across a variety of problems with better problem-solving reliability than the typical lognormal self-adaptation scheme. Examination of results from the lognormal self-adaptation scheme showed that best results came from different values of the $\tau$ parameter depending on whether the problem had unimodal or multimodal characteristics. Multimodal results suggest tentatively that the relationship presented in (3) holds for GAs. Furthermore, we found that the best results on all the multimodal problems came from using (3) with a setting of $c$ = 0.1.

A major finding of this work concerns the choice of an appropriate innovation rate for the self-adaptation mechanism. We found that although a low innovation rate provides the best performance for unimodal problems, in nearly all cases an innovation rate of one gives the best overall reliability and time to optimum figures for landscapes that have local optima or plateaus. Our hypothesis for the reasons for the benefits of a high innovation rate hinge on the relationship between selection pressure and the need for variety across strategy parameters in order for self-adaptation to function effectively. High selection pressure and self-adaptation algorithms favoring small step sizes have the effect of reducing population diversity and encouraging premature convergence. In this situation, self-adaptation schemes need to provide a means of stimulating diversity to counter the forces of selection. This is a side effect of high innovation rates, but it may be possible to design parameterless self-adaptation schemes that deliver this benefit directly, whilst still providing the advantages of traditional low innovation rate schemes, namely the transmission of historically successful strategies when they are appropriate. A scheme having these features was outlined and it was shown that the discrete model with uniform stochastic innovation provides these characteristics to a certain extent. Further experimentation with purpose-designed algorithms along these lines is needed to determine whether any further performance benefits can accrue. In addition, we need to test the strategy parameter diversity hypothesis on other problems and self-adaptation schemes.

### References


Bäck, T. (1992). Self adaptation in genetic algorithms. In F. Varela & P. Bourgine, (eds.), *Towards a Practice on Autonomous Systems: Proceedings of the First European Conference on Articial Life*, Cambridge, MA: MIT Press, pp. 263-271.

Bäck, T. & Hoffmeister, F. (1991). Extended selection mechanisms in genetic algorithms. In R. K. Belew & L. B. Booker, (eds.), *Proceedings of the Fourth International Conference on Genetic Algorithms and their Applications,* San Mateo, CA: Morgan Kaufmann, pp. 92-99.

Bäck, T. & Schütz, M. (1996). Intelligent mutation rate control in canonical genetic algorithms. In Z. Ras, (ed.), *Proceedings of the Ninth International Symposium on Methodologies for Intelligent Systems, Lecture Notes in Computer Science,* Berlin: Springer, pp. 158-167.

Baker, J. E. (1987). Reducing bias and innefficiency in the selection algorithm. In J. J. Grefenstette (ed.), *Genetic Algorithms and their Applications: Proceedings of the Second International Conference on Genetic Algorithms,* Hillsdale, NJ: Lawrence Erlbaum Associates, pp. 14-21.

Fogel, D. B., Fogel, L. J. & Atmar, J. W. (1991). Meta-evolutionary programming. In R. R. Chen (ed.), *Proceedings of the 25th Asilomar Conference on Signals, Systems and Computers*, San Jose, CA: Maple Press, pp. 540-545.

Glickman, M., R. & Sycara, K. (1998). Evolutionary algorithms: Exploring the dynamics of self-adaptation. In J. R. Koza et al (eds.), *Genetic Programming 1998: Proceedings of the Third Annual Conference,* San Francisco, CA: Morgan Kaufmann.

Glickman, M. R & Sycara, K. (2000). Reasons for premature convergence of self-adapting mutation rates. In *CEC2000: Proceedings of the Congress on Evolutionary Computation,* Piscataway, NJ: IEEE Press.

Hinterding, R. (1997). Self-adaptation using multi-chromosomes. *In Proceedings of the Fourth IEEE International Conference on Evolutionary Computation,* Piscataway, NJ: IEEE Press, pp. 87-91.

Hinterding, R., Michalewicz, Z. & Peachey, T. C. (1996). Self-adaptive genetic algorithm for numeric functions. In H-M. Voigt, W. Ebeling, I. Rechenberg, & H-P. Schwefel, (eds.), *Parallel Problem Solving from Nature - PPSV IV, volume 1141 of Lecture Notes in Computer Science*, Berlin: Springer, pp. 420-429.

Liang, K-H., Yao, X., Liu, Y., Newton, C. & Hoffman, D. (1998). An experimental investigation of self-adaptation in evolutionary programming. In V. Porto, N. Saravanan, D. Waagen & A. Eiben (eds.), *Evolutionary Programming VII: Proceedings of the Seventh Annual Conference on Evolutionary Programming, volume 1447 of Lecture Notes in Computer Science,* Berlin: Springer, pp. 291-300.

Rudolph, G. (1999). Self-adaptive mutations may lead to premature convergence. Technical Report CI-73/99, Department of Computer Science/XI, University of Dortmund.

Schwefel, H-P. (1997). Evolutionary computation - a study on collective learning. In N. Callaos, C. M. Khoong, E. Cohen (eds.), *Proceedings of the World Multiconference on Systemics, Cybernetics and Informatics, volume. 2,* Orlando FL: International Institute of Informatics and Systemics, pp. 198-205.

Schwefel, H-P. (1981). *Numerical Optimisation of Computer Models*. New York, NY: Wiley.

Smith, J. E. (2001). Modelling GAs with self adaptive mutation rates. In L. E. Spector et al (eds.), *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO-2001,* San Francisco, CA: Morgan Kaufmann, pp. 599-606.

Smith, J. E. & Fogarty, T. C. (1996). Self adaptation of mutation rates in a steady state genetic algorithm. In *Proceedings of IEEE International Conference on Evolutionary Computing 1996,* Piscataway, NJ: IEEE Press, pp. 318-323.

Stephens C. R., García Olmedo I., Mora Vargas J. & Waelbroeck, H. (1998). Self-Adaptation in Evolving Systems. *Artificial Life* 4(2), 183-201.

Stone, C. N. (2001). *Discrete Self-Adaptive Mutation in Genetic Algorithms.* M.Sc. dissertation. University of the West of England.

Yao, X., Lin, G. & Liu, Y. (1997). An analysis of evolutionary algorithms based on neighbourhood and step sizes. In P. Angeline, R. Reynolds, J. McDonnell & R. Eberhart (eds.), *Evolutionary Programming VI: Proceedings of the Sixth Annual Conference on Evolutionary Programming, volume 1213 of Lecture Notes in Computer Science*, Berlin: Springer, pp. 297-307.