Evolutionary Algorithms: Exploring the Dynamics of Self-Adaptation

Matthew Glickman and Katia Sycara
School of Computer Science
Carnegie Mellon University
5000 Forbes Ave.
Pittsburgh, PA 15381-3891
glickman+@cs.cmu.edu, katia+@ri.cmu.edu

ABSTRACT

Self-adaptation refers to allowing characteristics of search–most often mutation rates–to evolve on a per-individual basis rather than be specified by the user. This practice is gaining increasing attention and moving beyond classical mutation rates to explore other traits affecting search. The potential impact of self-adaptation is vast because it provides an implicit approach to problems of operator selection and parameter tuning, and possibly to those of representation as well. Studies have demonstrated many successful applications of self-adaptation, but in light of its potential impact, it is important to gain insight into the dynamics of this process to guide further experimentation. To this end, we present here an illuminating relationship between the strength of selection pressure and the magnitude of self-adapting mutation rates, as well as an observation on when self-adapting mutation rates are most likely to be of greatest utility.

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. The performance of evolutionary algorithms is thus ultimately limited by the ability of the variation process to continue to generate new forms of yet higher fitness. It is because the variation process is a direct result of the action of the chosen search operators on the chosen representation that the relative lack of general principles for making these choices is a worrisome issue. *Self-adaption* (for a review see (Angeline 1995)) is the practice of encoding aspects of these choices along with each member of the population, allowing them to vary and (hopefully) adapt on a per-individual basis. Self-adaptation can thus free the user from having to make *some* possibly arbitrary implementation choices, providing some measure of relief to representation and operator selection dilemma.

By far the most common class of trait to be self-adapted is that of mutation rates (see (Schwefel 1981), (Fogel *et al.* 1991), (Bäck 1992*b*)). A recent paper (Saravanan and Fogel 1997) extended this notion to self-adapt the *choice* between sampling mutations from one of two different underlying distributions. Other examples have included crossover (*e.g.* (Angeline 1996)) and inversion (Chellapilla and Fogel 1997).

The clearest path to further, more powerful mechanisms of self-adaptation is via a deeper understanding of its dynamics. In considering self-adaptation, perhaps the most immediate question is: How does the selection process act upon a trait that isn't directly measured by the objective function? To explore this and related questions, we've conducted experiments with self-adapting mutation rates in a simple domain, and observed phenomena consistent with the experimental results in a simplified domain model.

2 Implementation

Mutational operators, as defined by Jones (Jones 1995), "modify each component of their input with a given, typically small, probability". In practice, such modifications take many forms, typically driven by the chosen representation, *e.g.*: flipping bits in bitstring representations favored by Genetic Algorithms; substituting symbols in the Lisp parse-trees of Genetic Programming; or the addition of Gaussian noise to the real-valued vector elements of Evolution Strategies and (often) Evolutionary Programming. The mutation rate is a parameter that specifies the frequency or magnitude of the mutational events.

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.

In contrast to more commonly encoded traits, these individually varying mutation rates are never expressed in the individual's phenotype, the aspect of an individual that is evaluated by the fitness function. Encoded mutation rates instead constitute an example of a pure *transmission-related trait*, a trait which affects parent-offspring transmission by influencing the probability distribution from which the given individual's offspring are sampled.

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.

A few guidelines for the application of self-adapting mutation rates have been asserted, *e.g.* the importance of extinctive selection (Bäck 1992*b*). However, their theoretical analysis has only just begun (Beyer 1995).

2.1 The domain: "One Max"

To simplify the mechanics as much as possible, we chose a problem domain that was as simple as possible, Ackley's "One Max" problem (Ackley 1987). Bäck had already reported some results with self-adapting mutation rates in this domain (Bäck 1992*a*), providing us with a opportunity for comparison as well as a starting point for our investigation.

In One-Max each individual, g_i, is a bit-string of some length, l, that is globally fixed for the problem:

$$g_i = < b_1, ..., b_l > \in \{0, 1\}^l$$

The fitness function, $f : \{0, 1\}^l \rightarrow \{0..l\}$, then simply maps a bitstring to the total number of bits set to one among its l bits:

$$f(< b_1, b_2, ..., b_l >) = \sum_{i=1}^{l} b_i$$

The more bits set to 1 in a string, the higher its fitness.

2.1.1 Variation

Variation was induced via mutation, whereby a bit-string, $< b_1, ..., b_l >$, is copied to yield another bit-string, $< b'_1, ..., b'_l >$ identical to the original string *except* that each bit is inverted with a probability p, the mutation rate:

$$\text{for } i = 1 \text{ ... } l, \; b'_i = \begin{cases} b_i & \text{if } X > p \\ 1 - b_i & \text{otherwise} \end{cases}$$

where X is a random variable sampled from a uniform distribution over $[0, 1]$.

Without variable mutation rates, the mutation rate, p, is set to some fixed, global value for the duration of the run. If variable mutation rates are to be used, then each individual, g_i, is extended by some additional number of bits, r. These additional bits are ignored by the fitness function, but are interpreted by the mutation operator to yield p_i, the mutation rate for g_i. At the suggestion of Bäck (personal communication), we encoded the mutation rate for the One-Max problem as a real (floating-point) value, as is characteristic of both Evolution Strategies and Evolutionary Programming.

The magnitude of the mutation of the mutation rates themselves is then determined by another parameter–a globally fixed one in this case–τ. τ 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, p_i, then becomes the new mutation rate, p'_i:

$$p'_i = p_i \cdot e^{\tau \cdot N(0,1)}$$

where $\tau \cdot N(0, 1)$ denotes a random value sampled from a Gaussian distribution with mean 0 and standard deviation τ.

This form of "mutation-rate-mutation" has some nice properties, including the fact that it varies mutation rates in proportion to their own size. This is important because it is often the case that it is the *magnitude* of the mutation rate that really matters (rather than its *exact* value). Whether a mutation rate is 0.001 or 0.1, *multiplying* it by 1.2 is likely to be a plausibly interesting variation, while adding *0.08* may or may not. At the same time, the median value by which to multiply the mutation rate is still 1, so we're neither more likely to mutate it up or down. Moreover, multiplying by an exponential also guarantees that resulting mutation probability will never be meaninglessly negative, nor zero, which could prematurely stop further variation.

2.1.2 Selection

In the selection process employed here, known as (μ, λ) selection in the Evolution Strategies literature, the population, of size λ, is sorted in order of increasing fitness. Each of the top μ individuals in the resulting ordering are then copied $\frac{\lambda}{\mu}$ times to yield the new population.

3 Experimental Observations

3.1 Selection and mutation

Figures 1 and 2 races the average progress, over several runs, of the population mean fitness and mean mutation rate respectively. The fact that the mutation rates *do* evolve (rather than diffusing randomly) indicates a correlation between an individual's mutation rate and the expected number of offspring it is assigned via the selection process. In general then at any given generation, those individuals favored by the selection process–the highest fitness individuals–must be likely to have mutation rates within some particular range. Note that this is a rather probabilistic effect: A inappropriate mutation rate can still produce a high-fitness individual, and vice versa, but this is the exception rather than the rule.

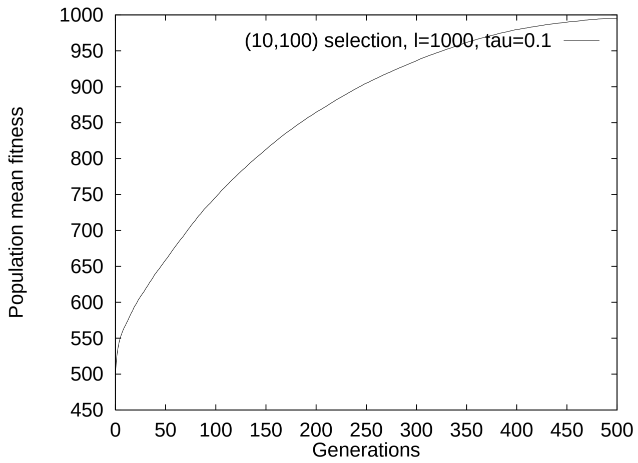

Figure 1 Population mean fitness over time, averaged over several runs

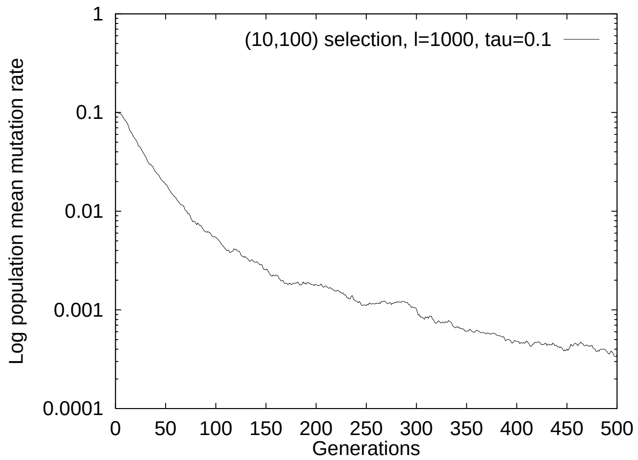

Figure 2 Population mean mutation rate over time, averaged over several runs

That lower mutation rates tend to be selected as fitness increases is not surprising in light of the fact that the higher the fitness, the more likely random perturbations are going to degrade it rather than improve it. The more specific issue of which particular mutation rate(s) are favored for individuals of a given fitness was addressed by Bäck (Bäck 1992*a*). He defined the optimal mutation rate for a string of given fitness as $p+$, 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.

Note, however, that 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. This possibility, however, is not borne out by experimental results (presented below), the reason for which is suggested by observation of a simplified model of this system, detailed in the following section.

3.2 A Two-Level Mutation Rate, Infinite Population Model

3.2.1 The Model

A model that permits clearer examination of the selective forces acting on self-adapting mutation rates in One-Max may be constructed via two simplifications. The first is to substitute the real-valued mutation rates with a simple, "two-level" self-adapting mutation rate[1] gene that only has two states: $p = 0.01$, and $p = 0.1$. We retain the global parameter τ, but it now simply refers to the probability of an individual's mutation rate changing from one value to the other. The dynamics resulting from this simplification may be significantly different than those of the original system over multiple generations. However, the property of providing for genetic variation over mutation rates in the population is still preserved, yielding the same essential interaction between this variability and the selection process in the short term.

The second simplification draws on Bäck's analysis (Bäck 1992*a*). of the probabilities of transitions in fitness from mutation in One-Max. Bäck's derivation of p_a^+, the probability that mutating a string with a 1's with mutation rate p will increase fitness, p_a^-, the probability that the mutation will decrease fitness, and p_a^0, the probability of no change in fitness, is easily extended to yield $p_a[b]$, the probability that mutation of an element with a 1's with mutation rate p will yield a string with fitness b. The model is shown in figure 3.

Given a particular fitness distribution, this analysis permits us to predict the expected change due to mutation over the entire population by considering in turn the number of individuals at each fitness level and their mutation rates. The results of mutation in finite populations will vary about these expected values due to sampling. However, the effects of sampling may be ignored by considering an "infinite" population, and instead speak of the frequencies in the population of each fitness value rather than numbers of individuals.

3.2.2 Observations: Mutation

Figures 4 and 5 show such a population (of strings of length l) before and after mutation respectively. In this and many of the following graphs, the range of fitness levels shown excludes the extrema where the frequencies of individuals are all not perceptibly greater than 0. The proportion of individuals at each fitness level that have each mutation rate is depicted via the two different levels of shading.

The first step in mutation is the mutation of the mutation rates. In general, in this model the effect of this step is to moderate the ratio of one mutation rate to the other. For example, if τ is, say, 0.2, and 90 percent of the population has

[1] Many thanks to Shumeet Baluja for suggesting this simplification

$$p_a[b] = \begin{cases} \sum_{i=b-a}^{l-a} \binom{l-a}{i} p^i (1-p)^{l-i} \binom{a}{i-(b-a)} p^i (1-p)^{l-i} & \text{if } b > a \text{ and } a+b \geq l \\ \sum_{i=b-a}^{b} \binom{l-a}{i} p^i (1-p)^{l-i} \binom{a}{i-(b-a)} p^i (1-p)^{l-i} & \text{if } b > a \text{ and } a+b < l \\ \sum_{i=a-b}^{l-b} \binom{l-a}{i-(a-b)} p^i (1-p)^{l-i} \binom{a}{i} p^i (1-p)^{l-i} & \text{if } b < a \text{ and } a+b \geq l \\ \sum_{i=a-b}^{a} \binom{l-a}{i-(a-b)} p^i (1-p)^{l-i} \binom{a}{i} p^i (1-p)^{l-i} & \text{if } b < a \text{ and } a+b < l \end{cases}$$

Figure 3 The probability that mutating an individual with a 1's with mutation rate p will yield a string with b 1's

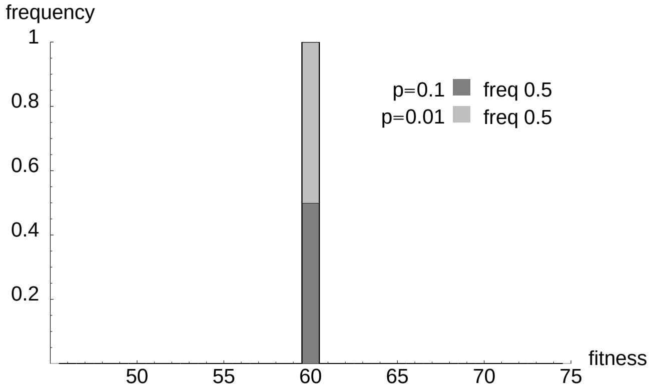

Figure 4 Initial frequency distribution

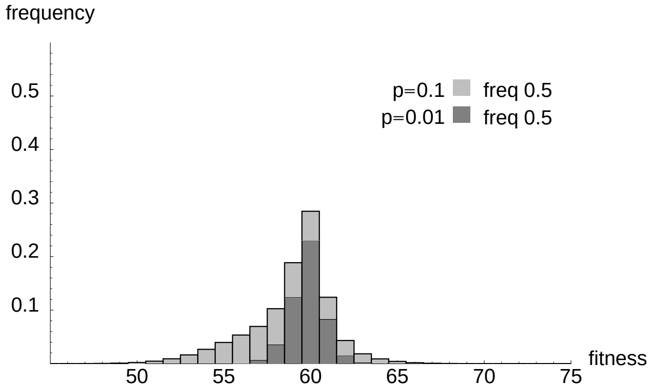

Figure 5 Frequency distribution after mutation

mutation rate 0.01 and the other 10 percent has rate 0.1, when mutation occurs, $0.2 \cdot 90$ percent of the individuals will change from rate 0.01 to rate 0.1, while only $0.2 \cdot 10$ percent of the individuals will change from rate 0.1 to rate 0.01. However, as is the case in figure 4, starting with an equal proportion of individuals with each mutation rate at each fitness level (in this case only 1), no change to this proportion is expected due to the mutation of the rates.

The mutation on the One-Max part of the strings is in a sense a moderating effect as well. As discussed in the previous section, when mutating a string with more 1's than 0's, more 1's are expected to change to 0's than 0's are expected to change to 1's. The net result is that mutation of such strings will *tend* to decrease their fitness, although strings of higher fitness will always be produced with some frequency.

As can be seen in figure 5, the distribution of fitness resulting from mutation at a rate of 0.1 is more spread out than that from mutation at 0.01. Moreover, the less conservative mutation rate results in more individuals of at both the highest (perceptible) fitness levels as well as the lowest, but because of mutation's general fitness-degrading tendency, the mean of the distribution of individuals resulting from the higher mutation rate is in fact less than that of the distribution resulting from the lower rate. Looking now at the effect of selection on the mutated distribution leads to insight into selection's action on mutation rates in general.

3.2.3 Observations: Selection

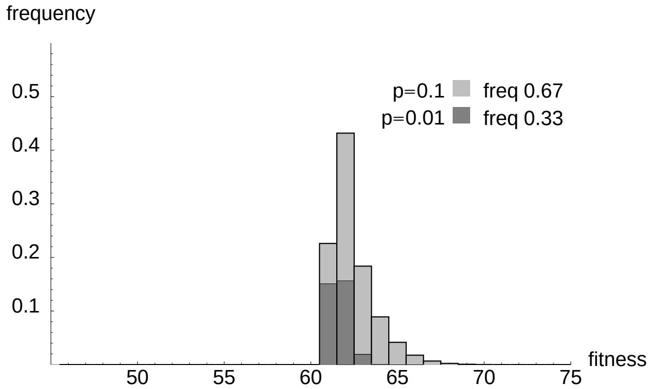

Figure 6 Frequency distribution after selecting the top tenth

The application (μ, λ) selection with $\frac{\mu}{\lambda} = 0.1$ simply "chops off" the fitness distribution below the 90th percentile, thus multiplying the remaining frequencies by 10. As it turns out, the proportion of rate 0.1 to rate 0.01 in the selected top 10 percent (see figure 6) of the mutated distribution (see figure 5) is actually near to two. Selection in this case thus favors the higher mutation rate. However, when selection is performed with $\frac{\mu}{\lambda} = 0.5$, the result is quite different (see figure 7). The top 50 percent of the population, as might be expected, contains more lower mutation rates than higher rates.

This example illustrates the general principle at work here. Given parents of better-than-random fitness, a smaller, more conservative mutation rate will yield an offspring fitness distribution that is both more narrow and possessed of a higher

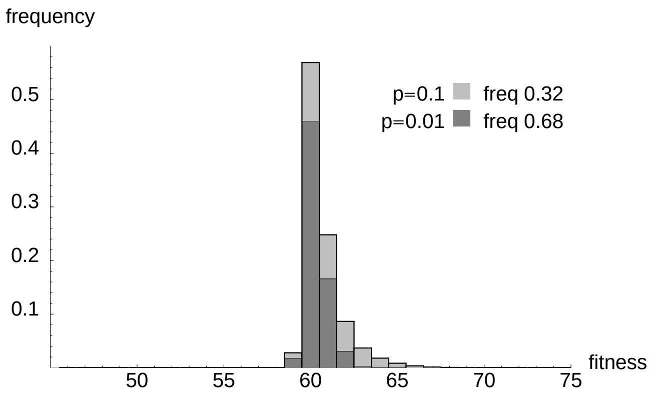

Figure 7 Frequency distribution after selecting the top half

mean that that of a larger, less conservative mutation rate. Although the lower-fitness tail of the distribution yielded by the higher mutation rate will grow more quickly than the higher-fitness tail, both will grow as the mutation rate increases. As the intensity of selection is increased, more offspring will be rewarded to an increasingly elite fraction of the fitness distribution, ultimately favoring the offspring of those individuals possessed of increasingly higher mutation rates.

3.2.4 Other Factors

Note that which specific mutation rates are favored to what degree in the population is also dependent on the pre-mutation parental fitness distribution. For example, 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; when you have little to lose, there is little incentive to “play it safe”.

Neither can we expect the mutation rates favored by selection to be independent of the shape of the particular fitness landscape. In One-Max, as the optimum is approached (the string of all ones, in the case of One-Max), there is less-and-less room on the high end of the fitness distribution for any “elite fraction” to be favored by stronger selection. Furthermore, real-world landscapes often have many local optima, a condition that is rarely negligible when considering evolutionary dynamics.

Nonetheless, One-Max captures one essential characteristic common to the vast majority of fitness landscapes: Points of increasingly high fitness are increasingly rare. One consequence of this fact is that the deleterious effect of mutation is usually more severe at higher fitness levels, which permits us to expect that independent of selection strength, selection will *tend* to favor individuals with lower mutation rates as fitness increases. Another consequence appears to be, judging from empirical observations (an example is shown below), that the relationship between selection strength and favored mutation rates demonstrated for One-Max generally carries over to other problem domains.

3.3 Selection strength and the favored mutation rate in the original system

Considering again the full evolutionary algorithm described in section 2, 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 (as in (Price 1970)). In (μ,λ)-selection, RS is determined by the number of offspring that rank in the top μ of the population. The mutation rate that should then be favored is the one which maximizes the expected number of offspring being of sufficiently high fitness to rank in this fraction. The higher the selection strength (the lower the ratio of μ to λ), the smaller this fraction will be, and thus the higher the fitness value required for offspring to survive.

However, to the extent to which expected fitness *correlates directly* with RS–as is the case with “fitness proportional selection”, whereby individuals are selected to reproduce with a probability linearly proportional to their fitness–we should indeed expect selection to generally favor the lowest mutation rate available (after $l/2$ bits are set to one), as was postulated to be the general case at the end of section 3

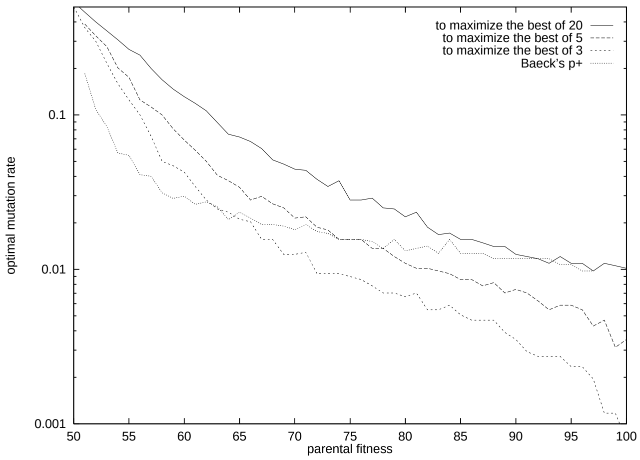

Figure 8 Mutation rates optimal for maximizing the best-out-of-n offspring

To estimate the mutation rates favored in (μ, λ) lambda selection, we ran large numbers of mutation trials at various rates in order to find the mutation rate that maximized the *highest* fitness produced from sets of n trials of the mutation of a parent bitstring. Parental fitness values were varied from 50 to 100, and n was set at each of 3, 5, and 20, the results of which are shown in figure 8 (including a similarly derived curve for Bäck’s $p+$). Each of the rate curves decreases with fitness, but the rate optimal to maximize the best-of-n

increases with n. To see how well these curves matched the actual rates selected during evolution, we wanted to know, for various selection strengths, which mutation rate was selected on average at each population mean fitness level (as opposed to at each generation).

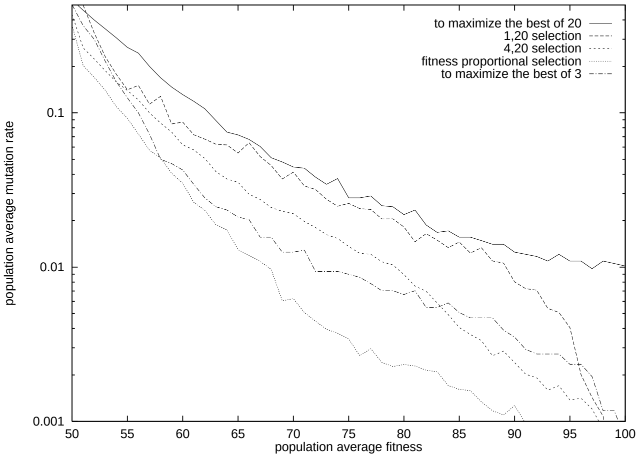

Figure 9 Average mutation rate by fitness for varying selection strengths

Due to the amount of variance in behavior between individual runs, it was necessary to collect data from a large number of One-Max runs (100) for each selection strength. The population average mutation rate was noted at each population average fitness level, and then averaged over all runs of a given strength. The results are shown in figure 9, along with some of the empirically derived "optimal" curves for comparison.

In general, the curves seem to match relatively well, *e.g.* the optimal rates to maximize the best of 20 mutations seem quite close to the population-mean mutation rates that are evolved over the course of $(1, 20)$ selection, in which each parent has 20 offspring, and is hence mutated 20 times. Note that, in line with the expectation raised above, the mutation rates decrease the *most* quickly with fitness in the case of fitness-proportional selection

4 An Alternate Problem Domain

We've found that the relationship between stronger selection and higher average mutation rates carries over both to at least one other fitness scheme (tournament selection) that does not assign offspring in a manner linearly proportional to fitness, as well as to other, more complex problem domains that have local optima. An example shown here is from an artificial neural network training problem originally described by Baluja (Baluja 1995), referred to here as the Box problem. The target function is a classification of points from a fixed region of the XY-plane. Points found inside a larger rectangular region but *not* inside a smaller, enclosed rectangle are to be labeled 0.5, while all others are to be labeled -0.5. Each individual in the evolving population encodes a vector of connection weights for a fixed architecture neural network.

The architecture is that of a feedforward network of sig-

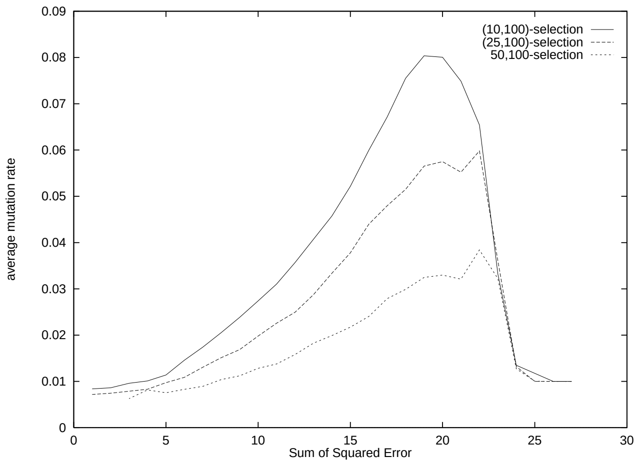

Figure 10 Population average mutation rate vs. Sum of Squared Error for the Box problem

moid units with five units in a single hidden layer, and a one unit in the output layer. A network's fitness measure is the sum of squared error values (with respect to the target function) accumulated as each of 100 uniformly distributed training examples are presented as input to the network. Each training example consists of a pair of X and Y coordinates, along with 5 separate random noise values that are chosen before the run and then remain fixed for each XY pair.

There are a total of 46 connection weights that are supplied by each population member. Each population members consists of a vector of 46 floating-point weight values, along with a corresponding vector of 46 floating-point mutation rates. Mutation of the weights is via the addition of Gaussian noise, and the standard-deviation of the distribution from which the mutational noise is sampled is determined via its own associated mutation rate.

Figure 10 shows the association between the population mean fitness (the sum of the squared error values, in this case) and the population mean mutation rate for each of three different (μ, λ) selection intensities. The data were collected from and averaged over several runs at each selection intensity. There are two distinct differences from the results presented for One-Max in figure 9: (1) The mutation rates were all initially set to a low value and were then quickly selected up to a peak value before beginning to descend with fitness; and (2) "better" fitness in this case corresponds to *lower* error values, explaining why rates are decreasing going to the left rather than to the right as in One Max.

Implementation details aside, the key point is that we find the same correlation between the chosen selection intensity and the population mean mutation rate favored for each mean fitness level as shown above for One Max. This result is particularly indicative of the generality of this effect given the differences between the Box problem scenario presented here and that of One Max. Differences in this case include: 46 real-valued traits per individual, rather than 100 binary traits; mutation via the addition of Gaussian noise, rather than bit-flipping; the specification of variable mutation rates on a one-per-trait basis, rather than simply one rate per individual; and a much more complicated fitness landscape containing many local optima. Although pathological cases may well exist where the general relationship between selection strength and variable mutation rates is absent or fundamentally different, we have yet to observe them despite experience with a variety of differing scenarios.

5 When are self-adapting rates most beneficial?

Insight into the dynamics of self-adaptation does not necessarily address questions of utility, such as when the use of self-adapting rates is likely to be the most advantageous for search.

In fact, for the One Max domain our findings were in fact that the use of a single, well-chosen, fixed global mutation rate often yielded equal or better results than the use of one-per-individual self-adapting rates. For practical purposes, however, this result says very little because in fact, the number of trials it took to find an ideally well-adapted fixed rate typically *far* exceeded the amount of time it took to conduct a self-adapting run.

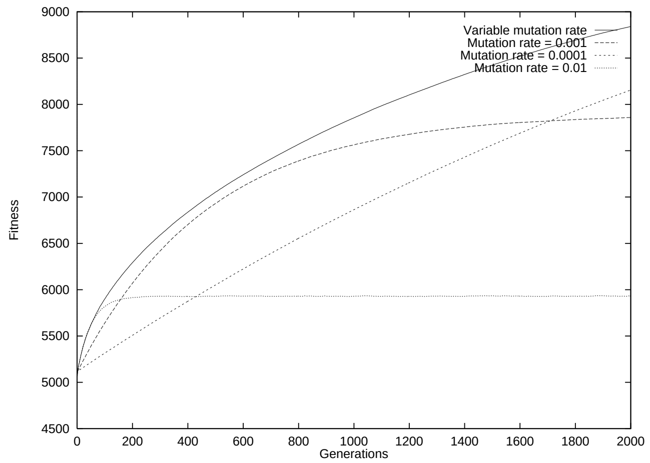

Figure 11 Fitness over time for fixed vs. variable mutation rates with $l = 10000$

One situation in which we observed self-adapting mutation rates become clearly advantageous is *once the string is long enough* (see figure 11). This result seems likely to be due to the very large discrepancy in magnitude of the mutation rates that produce reasonable progress near the beginning, and those that produce reasonable progress near the end when evolving a very long string.

Because with shorter strings fixed-rates often outperform variable rates, it would seem that there is indeed a cost to self-adaptation. At the same time, the example presented here suggests one point where the benefit of self-adaption exceeds its cost: when there is a sufficient difference between the modes of variation (*i.e.* parent-offspring transmission) necessary to make reasonable progress in differing regions of the search space that must be traversed to arrive at the goal.

6 Conclusion

We've offered some insight into the relationship between selection and self-adapting mutation rates. It seems likely that the relationship between selection intensity and variable mutation rates explored here may generalize beyond just mutation rates. This is because the understanding of this effect gleaned in section 3.2 seems applicable to any transmission-related trait that varies the degree of conservatism of parent-offspring transmission. This hypothesis is in line with results presented on the action of selection on representation in Genetic Programming (Altenberg 1994), particularly in the case of the "program bloat" phenomenon of Genetic Programming (*e.g.* (Soule and Foster 1997), (Blickle and Thiele 1994)).

We are currently exploring this and related hypotheses about the dynamics of self-adaptation. The more insight that is gained into the dynamics underlying self-adaptation, the more likely it is that self-adaptation can be further extended to provide for successful search in problem domains currently beyond reach of exploration.

7 Acknowledgements

The authors wish to thank James Thomas for many extended, helpful discussions, Shumeet Baluja for suggestions and encouragement, and two anonymous reviewers for their thoughtful commentary and corrections. This research was partially supported by the Office of Naval Research, Contract N00014-95-1-0591. Matthew Glickman was partially supported by an NSF Graduate Fellowship.

References

Ackley, D. H. (1987). *A connectionist machine for genetic hillclimbing*. Kluwer. Boston.

Altenberg, Lee (1994). The evolution of evolvability in genetic programming. In: *Advances in Genetic Programming* (K. Kinnear, Ed.). MIT Press.

Angeline, P. J. (1995). Adaptive and self-adaptive evolutionary computations. In: *Computational Intelligence: A Dynamic Systems Perspective* (M. Palaniswami, Y. Attikiouzel, R. Marks, D. Fogel and T. Fukuda, Eds.). pp. 152–163. IEEE Press. Piscataway, NJ.

Angeline, P. J. (1996). Two self-adaptive crossover operations for genetic programming. In: *Advances in Genetic Programming II* (P. Angeline and K.Kinnear, Eds.). pp. 152–163. MIT Press.

Angeline, P.J., Reynolds, R.G., McDonnell, J.R. and Eberhart, R., Eds.) (1997). *Evolutionary Programming VI*. Springer.

Bäck, T. (1992*a*). The interaction of mutation rate, selection, and self-adaptation within a genetic algorithm. In: *Parallel Problem Solving 2* (R. Männer and B. Manderick, Eds.). Elsevier. Amsterdam.

Bäck, Thomas (1992*b*). Self-adaptation in genetic algorithms. In: (Varela and Bourgine 1992). pp. 263–271.

Baluja, Shumeet (1995). An empirical comparison of seven iterative and evolutionary function optimization heuristics. Technical Report CMU-CS-95-193. School of Computer Science, Carnegie Mellon University.

Beyer, Hans-Georg (1995). Towards a theory of evolution strategies. *Evolutionary Computation* **3**(3), 311–347.

Blickle, Tobias and Lothar Thiele (1994). Genetic programming and redundancy. In: *Genetic Algorithms within the Framework of Evolutionary Computation, Proceedings of the KI-94 Workshop*. number MPI-I-94-241.

Chellapilla, K. and D. B. Fogel (1997). Exploring self-adaptive methods to improve the efficiency of generating approximate solutions to traveling salesman problems using evolutionary programming. In: Angeline *et al.* (1997).

Fogel, D. B., L. J. Fogel and J. W. Atmar (1991). Meta-evolutionary programming. In: *Proceedings of 25th Asilomar Conference on Signals, Systems and Computers* (R. R. Chen, Ed.). Pacific Grove, California. pp. 540–545.

Jones, Terry (1995). Evolutionary Algorithms, Fitness Landscapes, and Search. PhD thesis. The Univerisity of New Mexico. Albuquerque, New Mexico.

Price, G. R. (1970). Selection and covariance. *Nature* **227**, 520–521.

Saravanan, N. and D. B. Fogel (1997). Multi-operator evolutionary programming: A preliminary study on function optimization. In: Angeline *et al.* (1997).

Schwefel, H. P. (1981). *Numerical Optimization of Computer Models*. Wiley. Chichester.

Soule, Terence and James A. Foster (1997). Code size and depth flows in genetic programming. In: *Genetic Programming 1997* (J. R. Koza, K. Deb, M. Dorigo, D. B. Fogel, M. Garzon, H. Iba and R. L. Riolo, Eds.). Morgan Kauffman. San Francisco.

Varela, Francisco J. and Bourgine, Paul, Eds.) (1992). *Towards a Pratice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life*. MIT Press. Cambridge, MA.