On Recombination and Optimal Mutation Rates

Gabriela Ochoa, Inman Harvey, Hilary Buxton
Centre for Computational Neuroscience and Robotics
School of Cognitive and Computing Sciences
The University of Sussex
Falmer, Brighton BN1 9QH, UK

Abstract

We present empirical evidence, from a wide range of problem characteristics, suggesting that the value of optimal mutation rates in GAs differs according to whether recombination is used or not. Without recombination, a regime that starts with a high mutation rate, decreasing it towards the end of the run, appears to be optimal. With recombination, however, the optimal strategy proves to be a constant, sufficiently low mutation rate. Moreover, when recombination is used, the choice of an excessively high mutation rate might degrade the algorithm's performance considerably. These results are supported by recent knowledge from the field of molecular evolution about the effect of recombination on the so called *error thresholds*. We conclude by proposing a novel argument favoring the use of recombination in GAs. This argument, which we call the *dual-role* of recombination, sheds new light on the role of this operator in genetic search.

1 INTRODUCTION

It has long been acknowledged that a GA's performance depends heavily on the choice of its main parameters: mutation rate, crossover rate, and population size. These parameters typically interact with one another in a nonlinear fashion, so they cannot be independently optimized. Optimal parameter settings have been the subject of numerous studies in the GA literature, but there is no conclusive agreement on what is best; most people use what has worked well in previously reported cases.

Particular emphasis has been placed on finding optimal mutation rates [Fogarty, 1989, Mühlenbein, 1992, Hesser and Männer, 1991, Bäck, 1993]. Most theoretical studies aimed at finding optimal mutation values, however, neglect recombination in order to simplify the analysis [Bäck, 1993, Hesser and Männer, 1991, Mühlenbein, 1992]. On the other hand, classical empirical studies aimed at finding optimal parameter settings, use a fixed set of test problems [DeJong, 1975, Grefenstette, 1986, Schaffer et al., 1989]. One weakness with these classical studies is that their results may not generalize beyond the test problems used.

According to Spears [Spears, 1998] there are two ways to strengthen the results obtained from empirical studies. The first is to remove the opportunity to hand-tune algorithms to a particular set of problems. The second is to always show results over the running time of a GA (see section 3). In this paper we use these methodological guidelines to show that the choice of an optimal mutation scheme depends on whether recombination is used or not. For a GA without recombination, the optimal strategy appears to be the generally acknowledged heuristic of starting with a relatively high mutation rate, reducing it over the course of a single run [Fogarty, 1989, Mühlenbein, 1992, Bäck, 1991, Bäck, 1993]. However, when recombination is used, a fixed, sufficiently low mutation rate proves to be the optimal strategy. Moreover, with recombination, the GA performance is more sensitive to the use of an inappropriately high mutation rate.

These are more than just empirical results: theoretical knowledge from the field of molecular evolution support them. The argument, explained in more detail in section 2, is that the notion of optimal mutation rates is related to the so called "error thresholds". And thus the effects of recombination on error thresholds are reflected on optimal mutation rates.

This explanation and further insight, lead us to proposing a new argument favoring the use of recombination in Evolutionary Algorithms (EAs). This argument, which we call the *dual-role* of recombination, help us in understanding the role of this complex operation in EAs (see section 5).

In the reminder of the paper we summarize the knowledge from molecular evolution relevant to our argument, we describe the empirical methodology used, we present the experimental results obtained, and we discuss the insight gained.

2 ERROR THRESHOLDS

The error threshold — a notion from molecular evolution — is a critical mutation rate beyond which structures obtained by the evolutionary process are destroyed more frequently than selection can reproduce them. With mutation rates above this critical value, an optimal solution would not be stable in the population, i.e., the probability that the population loses these structures is not negligible.

The notion of error threshold, then, seems to be intuitively related to the idea of an optimal balance between *exploitation* and *exploration* in genetic search. Too low a mutation rate implies too little exploration; in the limit of zero mutation, successive generations of selection remove all variety from the population, and once the population has converged to a single point in genotype space all further exploration ceases. On the other hand, clearly, mutation rates can be too excessive; in the limit where mutation places a randomly chosen allele at every locus on an offspring genotype, then the evolutionary process has degenerated into random search with no exploitation of the information acquired in preceding generations.

Any optimal mutation rate must lie between these two extremes, but its precise position will depend on a number of factors including, in particular, the structure of the fitness landscape under consideration. It can, however, be hypothesized that a mutation rate just below the error threshold is the optimal mutation rate for the landscape under study. The close correspondence between error thresholds and optimal mutation rates may be assessed empirically. Given that mutation rates should not be above error thresholds, it cannot be immediately assumed that optimal mutation rates are related to this upper bound; however, experiments where the error threshold and the optimal mutation rates could be assessed independently showed that there was such a relationship. These experiments will be reported in detail elsewhere.

Some biological evidence supports the relationship between error thresholds and optimal mutation rates. Eigen and Schuster [Eigen and Schuster, 1979] have pointed out that viruses — which are very efficiently evolving entities — live within and close to the error thresholds given by the known rates of nucleotide mutations. This correspondence has also been noticed before in the GA community: Hesser and Manner [Hesser and Männer, 1991], devised an heuristic formula for optimal setting of mutation rates inspired by Nowak and Schuster's work on error thresholds [Nowak and Schuster, 1989]. Moreover, Kauffman [Kauffman, 1993] (p. 107), talking about an optimum mutation rate, suggests that "That rate is likely to occur when populations are just beginning to melt from peaks".

2.1 RECOMBINATION AND ERROR THRESHOLDS

A relatively recent work from the evolutionary biology literature [Boerlijst et al., 1996], reports interesting results about the role of recombination on evolving population of viruses. In particular, they study the effect of recombination on the magnitude of the error threshold. A mathematical model with infinite populations was used. Their results may be summarized as follows: for low mutation rates, recombination can focus the population around a fitness optimum and thus enhance overall fitness. For high mutation rates, however, recombination can push the population over the error threshold, and therefore cause a loss of genetic information. In other words, recombination shifts the error threshold to lower mutation rates, and, in addition, makes this transition sharper. The explanation given by the authors to this phenomenon is as follows [Boerlijst et al., 1996] (p. 1581):

> Near the error threshold, without recombination, the fittest strain only makes up a small percentage of the total population [Eigen and Schuster, 1979]. Under such conditions recombination acts as a diverging operation, driving the population beyond the error threshold. There can be selection for recombination if fitness is correlated and if the mutation rate is sufficiently small.

In [Ochoa and Harvey, 1998] we reproduce, using GAs — and hence finite populations — some of the results obtained by Boerlijst et al. GA simulation results were strikingly similar qualitatively to those obtained analytically. Thus, the main results described above for infinite populations also hold for an evolving (finite) population of bit-strings using a standard GA.

3 METHODS

Recently, De Jong, Spears, and Potter proposed a new empirical methodology for studying the behaviour of EAs [DeJong et al., 1997, Spears, 1998]. This approach employs the so called *problem generators.* A problem generator is an abstract model capable of producing randomly generated problems on demand. The advantages of using problem generators are two-fold. First, they allow us to report results over a randomly generated set of problems rather than a few hand-chosen examples, increasing in this way the predictive power of the results for the problem class as a whole. Secondly, problem generators are quite easy to parameterize, allowing the design of controlled experiments where particular features of a class of problems can be varied systematically to study the effects on the EA behavior.

For our study, we adopted this methodology and selected two problems generators: (i) the NK-Landscape generator (section 3.1), and (ii) the Multimodal generator (section 3.2).

3.1 THE NK-LANDSCAPE GENERATOR

Kauffman [Kauffman, 1989], describes a family of fitness landscapes determined by two parameters: N and K. The points of the NK-Landscape are bit strings of length N. The parameter K represents the *degree of epistatic interaction* between the bits, that is, the number of linkages each locus has to other loci in the same string. To compute the fitness of the entire string s, the fitness contribution from each locus is averaged as follows:

$f(s) = \frac{1}{N}\sum_{i=1}^{N} f(locus_i)$,

where the fitness contribution of each locus, $f(locus_i)$, is determined by using the (binary) value of gene i together with values of the K interacting loci as an index into a table T_i of size 2^{k+1} of uniformly distributed random numbers over $[0.0, 1.0]$. For a given locus i, the set of K linked loci may be randomly selected or consist of the immediately adjacent loci.

An interesting property of the NK-landscapes is that the ruggedness of the fitness landscape can be tuned by changing the parameter K. From a practical perspective, however, the NK-landscape presents some difficulties (in particular the large space required to store the tables to compute the fitness) which restrict its use to relative small models.

3.2 THE MULTIMODAL GENERATOR

The multimodal generator was proposed recently by De Jong, Potter, and Spears [DeJong et al., 1997]. The idea is to generate P random N-bit strings, which represent the location of the P peaks in the space. To evaluate any bit string s, first locate the nearest peak (in Hamming space). Then the fitness of s is the number of bits s has in common with that nearest peak, divided by N.

$f(s) = \frac{1}{N} max_{i=1}^{P}(N - Hamming(s, Peak_i))$

Problems with a small/large number of peaks are weakly/strongly epistatic. The multimodal generator is very efficient in terms of memory storage (only the P peaks need to be stored). However, the computation of fitness becomes very slow as the number of peaks is increased.

4 EXPERIMENTAL RESULTS

Following the guidelines of De Jong et al., the experimental methodology used was as follows [DeJong et al., 1997]: for each of the selected settings of the problem generator parameters, 20 problems were randomly generated. The GA was run once per problem, and the results were averaged over those 20 problems.

For all the experiments, a standard generational GA with fitness proportional selection was employed. Population size and chromosome length were set to 100. Two-point crossover and the standard bit mutation operation were used. For the GA with recombination, a crossover rate of 0.6 was selected. These are quite typical settings for GAs. Experiments were run for a maximum of 1000 generations.

To see how the mutation rate value affects the GA performance with and without recombination, we selected three mutation rates (0.001, 0.005, and 0.01), and ran the algorithm in two modes. In the first mode (GA) both mutation and recombination were used. In the second mode (GA-m) only mutation was used. Table 1 summarizes the GA parameter setting used for the experiments.

The performance metric we monitored is well-known – namely "best-so-far" curves that plot the fitness of the best individual that has been seen thus far by generation n. Each curve plots the average best-so-far values of 20 runs. For the sake of clarity, the standard deviations for these curves were not plotted. However, they all showed to be quite low — in the range of [0.01, 0.02].

Chromosome length	100
Population size	100
Crossover rate	0.6 (GA), 0.0 (GA-m)
Mutation rate	0.001, 0.005, 0.01
Generations	1000
No. of Problems	20

Table 1: GA parameters

4.1 NK EXPERIMENTS [1]

Given that we selected relatively long chromosomes, the storage requirements for the NK tables make it difficult to explore large values of K. Thus, we tested NK landscapes for $K = 0$, and $K = 2$. For more complex landscapes we relied on the multimodal problem generator results (section 4.2). The NK model with $K = 0$, produces a very trivial "Mount Fuji" landscape. We used it, however, as a baseline comparison before moving on to more interesting landscapes.

Figures 1 and 2 illustrate results for GA and GA-m on the NK landscape with K equals zero.

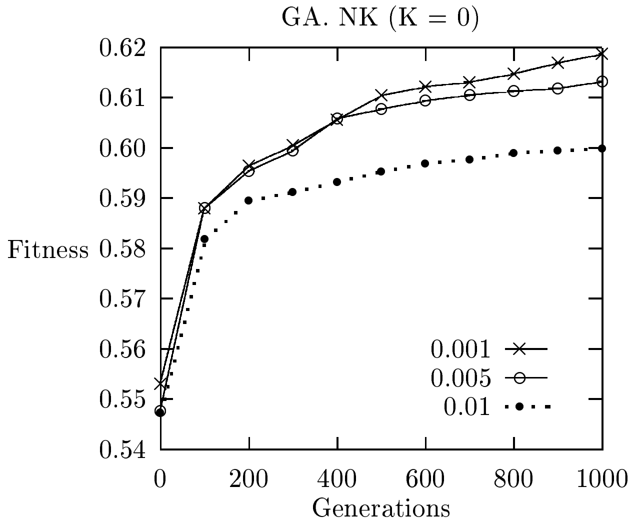

Figure 1: Average best-so-far curves. GA with distinct mutation rates on the NK landscape ($K = 0$)

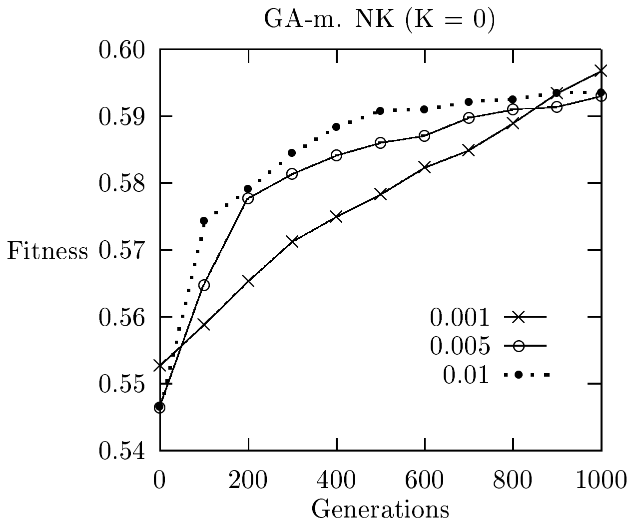

Figure 2: Average best-so-far curves. GA-m with distinct mutation rates on the NK landscape ($K = 0$)

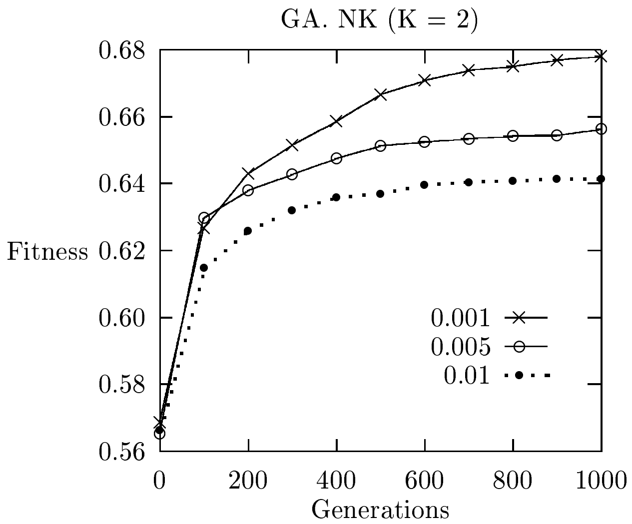

Figure 3: Average best-so-far curves. GA with distinct mutation rates on the NK landscape ($K = 2$)

Figures 3 and 4 show the average best-so-far curves for a level of epistasis K of two, with and without recombination.

When recombination is used, it can be clearly noticed that the lowest mutation rate explored (0.001) produces the best results over the entire algorithm run (Figures 1 and 3). This is more evident for the more epistatic landscape — when K equals two (Figure 3). On the other hand, for the GA without recombination (GA-m), the higher mutation rates (0.005 and 0.01) speed up noticeably the search process at the beginning and intermediate stages of the search (Figures 2 and 4), however, by the final stages of the run the lowest mutation rate curve (0.001) approaches the other

[1] For the NK experiments, we used the freeware implementation due to M. Potter

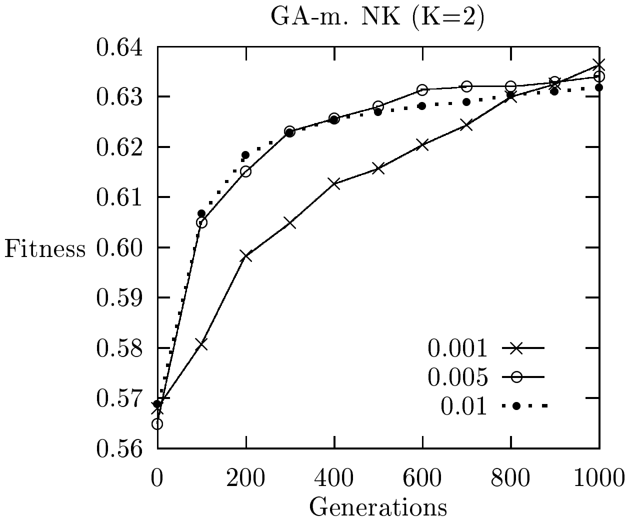

Figure 4: Average best-so-far curves. GA-m with distinct mutation rates on the NK landscape ($K = 2$)

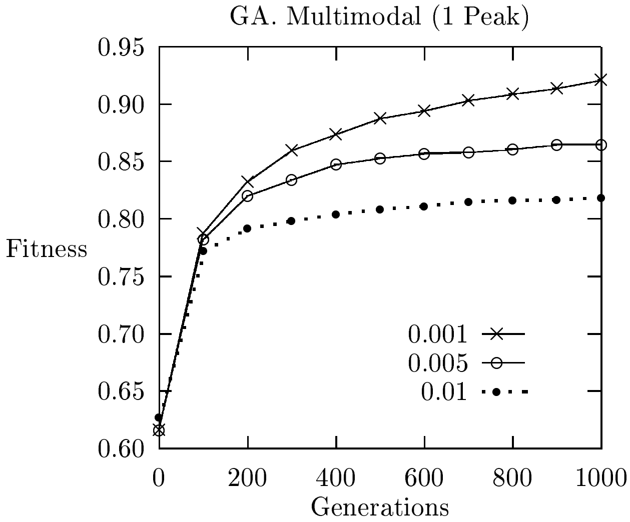

Figure 5: Average best-so-far curves. GA with distinct mutation rates on the multimodal landscape (1 peak)

two, and finally reaches the highest fitness values.

4.2 MULTIMODAL GENERATOR EXPERIMENTS [2]

Experiments were run for 1, 100, and 500 peaks problems. Figures 5 and 6 show the average best-so-far curves for a GA with and without recombination on 1 peak problems. Figures 7 and 8 show the average best-so-far curves for for GA and GA-m on 100 peaks problems, whereas Figures 9 and 10 do so on 500 peaks problems.

Again, when recombination is used, the lowest mutation rate explored (0.001) produces the best performance over the entire algorithm run for 1, 100 and 500 peaks problems (Figures 5, 7, and 9). Moreover, it can be clearly seen that while increasing the number of peaks, the effect is more pronounced. In other words, the difference between the best-so-far curves is more noticeable.

Without recombination, again, the higher mutation rates explored (0.005 and 0.01) increased performance at early stages (Figures 6, 8 and 10). Note, however, that eventually the performance curves for the lowest mutation rate (0.001) pick up in later generations. This occurs earlier for the more complex landscapes (those with 100 and 500 peaks — Figures 8 and 10). What appears to be happening is that at later stages

[2] For the multimodal generator experiments, we used the freeware implementation due to W. Spears

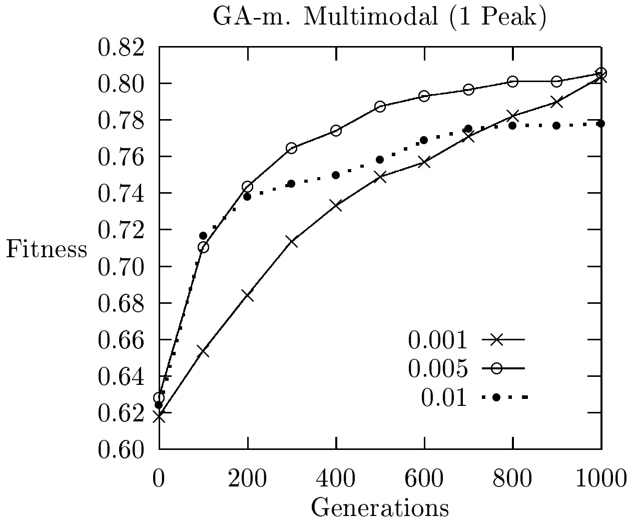

Figure 6: Average best-so-far curves. GA-m with distinct mutation rates on the multimodal landscape (1 peak)

of the search, only a few bits need to be changed, and a high mutation rate might have a disruptive effect.

5 DISCUSSION

In this paper we used the so-called problem generators to empirically explore optimal mutation rates for GAs with and without recombination. The main conclusion

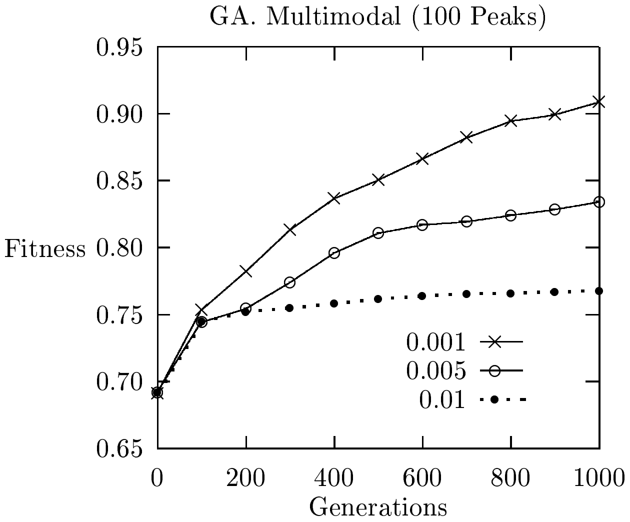

Figure 7: Average best-so-far curves. GA with distinct mutation rates on the multimodal landscape (100 peaks)

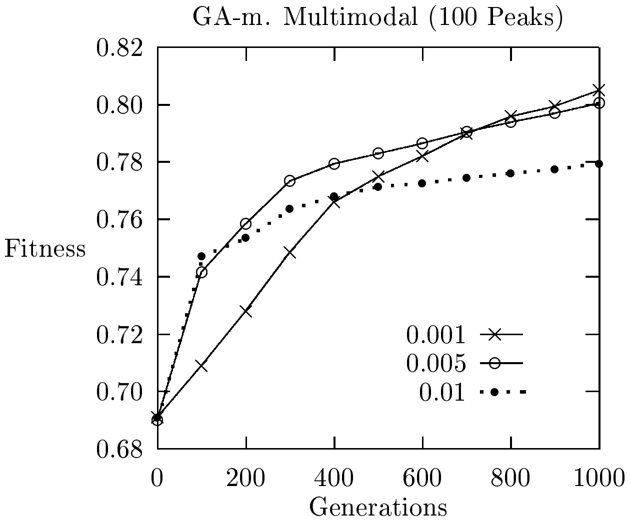

Figure 8: Average best-so-far curves. GA-m with distinct mutation rates on the multimodal landscape (100 peaks)

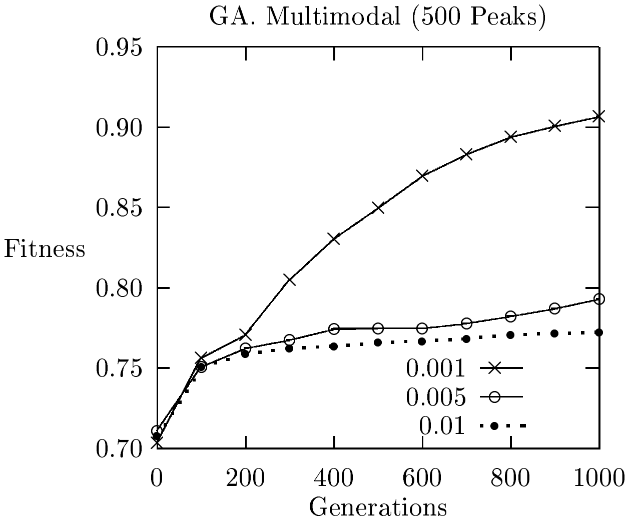

Figure 9: Average best-so-far curves. GA with distinct mutation rates on the multimodal landscape (500 peaks)

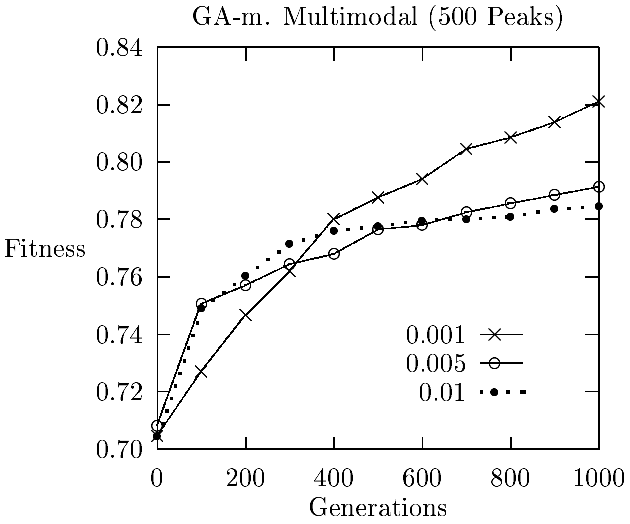

Figure 10: Average best-so-far curves. GA-m with distinct mutation rates on the multimodal landscape (500 peaks)

holds for all the scenarios studied: the optimal mutation scheme for a genetic algorithm differs according to whether recombination is used or not. For a GA with mutation and selection only, the search process benefits from starting with a relatively high mutation rate, decreasing it towards the final stages of the search. These results are in agreement with previous observations reported in the literature that a time-dependent variation of the mutation rate may improve GA performance [Fogarty, 1989, Mühlenbein, 1992, Bäck, 1991, Bäck, 1993].

On the other hand, when recombination is used, a constant, relatively low mutation rate seems to be the optimal strategy. In this case, selecting an excessively high mutation rate — over the error threshold — considerably degrades the algorithms performance. This effect was shown to be more pronounced for more "complex" landscapes (i.e. with higher levels of epistasis or multimodality or both).

The proposed explanation for the observed effect of recombination on optimal mutation rates, is as follows: the notion of optimal mutation rates is related to the notion of error thresholds. Thus, the effects of recombination on error thresholds, described in some detail in section 2.1, occurs as well on optimal mutation rates.

Here, we highlight our interpretation of the observed results, which proposes a novel argument about the important role of recombination in EAs. Recombination performs a *dual-role* in genetic search according to the level of genetic convergence of the population. At the beginning of the algorithm's run, when the population is scattered over the search space, recombination acts as a diverging operation, thus increasing the algorithm's search power and speeding up the process. In this role, it can be said that recombination acts as a sort of "macro-mutation" operator. Towards the final stages of the search, however, when the population is more genetically homogeneous, recombination can focus the population around the fitness optimum. In this second role, recombination acts as an error repair mechanism, helping in getting rid of deleterious mutations. We conclude, then, that there is no need of implementing a time-dependent mutation regime when recombination is used: recombination implicitly does this job for us. This confers a great advantage and encourages, in our opinion, the use of recombination in EAs.

About the generality of these results, we must add that some other more traditional test functions were also investigated: the one-max function, the royal road function, and some functions from classical optimization test suites. Due to space limitations, we can only briefly state that similar results were obtained. We know, however, that despite all these efforts, it cannot be categorically assured that these results apply to all problem domains. It would be interesting to test these ideas on some real-world applications.

Another scenario where this ideas should be explored is on landscapes with *neutrality* (the extent to which distinct genotypes have the same or very similar fitness values) [Barnett, 1997]. The concept of error thresholds can be extended to such landscapes, and future work will investigate whether there is a similar correlation with optimal mutation rates in this scenario.

Preliminary experiments suggest that these results also hold for other crossover operators, such as one-point and uniform crossover. Higher crossover rates were also tested, and results suggest that the main conclusions not only hold but are more pronounced. What remains to be studied is the effect of changing both population size and chromosome length. We strongly believe that optimal mutation rates depend on the values of the above two parameters. However, the main conclusions presented here, most probably hold qualitatively.

In the light of these results, we propose two general heuristics for setting GA parameters:

- When recombination is used, the mutation rate must be sufficiently small and constant over the entire run.
- When recombination is not used, a regime that starts with a high mutation rate, decreasing it towards the end of the run, may accelerate the search process.

We argue that these heuristics have to be specially considered when empirically comparing the relative importance of mutation and recombination in genetic search. To be fair, comparisons should be made selecting the optimal mutation scheme for each strategy.

Some final words about methodology are worth mentioning. We strongly support the use of "test-problem generators" as an empirical methodology, due to its advantages mentioned above. In particular, we strongly agree that from both an engineering and scientific standpoint, it is crucial to consider the dynamic aspects of EAs by including results throughout their entire run.

Acknowledgements

Thanks are due to A. Meier and M. Sordo for help and support during this effort. Thanks to L. Mauro for valuable suggestions and critical reading. Thanks also to M. Potter and W. Spears for making their source code available through the Internet.

References

[Bäck, 1991] Bäck, T. (1991). Self-adaptation in genetic algorithms. In Varela, F. J. and Bourgine, P., editors, *Proceedings of the First European Conference on Artificial Life. Toward a Practice of Autonomous Systems*, pages 263–271, Paris, France. MIT Press, Cambridge, MA.

[Bäck, 1993] Bäck, T. (1993). Optimal mutation rates in genetic search. In Forrest, S., editor, *Proceedings of the 5th International Conference on Genetic Algorithms*, pages 2–8, San Mateo, CA, USA. Morgan Kaufmann.

[Barnett, 1997] Barnett, L. (1997). Tangled webs: Evolutionary dynamics on fitness landscapes with neutrality. Master's thesis, School of Cognitive and Computing Sciences.

[Boerlijst et al., 1996] Boerlijst, M. C., Bonhoeffer, S., and Nowak, M. A. (1996). Viral quasi-species and recombination. *Proc. R. Soc. London. B*, 263:1577–1584.

[DeJong, 1975] DeJong, K. A. (1975). *An Analysis of the Behavior of a Class of Genetic Adaptive Systems*. PhD thesis, University of Michigan, Ann Arbor, MI. Dissertation Abstracts International 36(10), 5140B, University Microfilms Number 76-9381.

[DeJong et al., 1997] DeJong, K. A., Potter, M. A., and Spears, W. M. (1997). Using problem generators to explore the effects of epistasis. In Bäck, T., editor, *Proceedings of the 7th International Conference on Genetic Algorithms*, pages 338–345, San Francisco. Morgan Kaufmann.

[Eigen and Schuster, 1979] Eigen, M. and Schuster, P. (1979). *The Hypercycle: A Principle of Natural Self-Organization*. Springer-Verlag.

[Fogarty, 1989] Fogarty, T. C. (1989). Varying the probability of mutation in the genetic algorithm. In Schaffer, J. D., editor, *Proceedings of the 3rd International Conference on Genetic Algorithms*, pages 104–109, George Mason University. Morgan Kaufmann.

[Grefenstette, 1986] Grefenstette, J. J. (1986). Optimisation of control parameters for genetic algorithms. *IEE Trans SMC*, 16(1):122–128.

[Hesser and Männer, 1991] Hesser, J. and Männer, R. (1991). Towards an optimal mutation probability for genetic algorithms. In Schwefel, H.-P. and Männer, R., editors, *Parallel Problem Solving from Nature*. Springer-Verlag, Lecture Notes in Computer Science Vol. 496.

[Kauffman, 1989] Kauffman, S. (1989). Adaptation on rugged fitness landscapes. In Stein, D., editor, *Lectures in the Sciences of Complexity*, pages 527–618. Addison-Wesley, Reading, MA.

[Kauffman, 1993] Kauffman, S. A. (1993). *The Origins of Order: Self-Organization and Selection in Evolution*. Oxford University Press.

[Mühlenbein, 1992] Mühlenbein, H. (1992). How genetic algorithms really work: I. mutation and hillclimbing. In Männer, B. and Manderick, R., editors, *Parallel Problem Solving from Nature, 2: Proceedings of the Second Conference on Parallel Problem Solving from nature, Brussels*, pages 15–25. North-Holland.

[Nowak and Schuster, 1989] Nowak, M. and Schuster, P. (1989). Error thresholds of replication in finite populations: Mutation frequencies and the onset of Muller's ratchet. *J. Theor. Biol.*, 137:375–395.

[Ochoa and Harvey, 1998] Ochoa, G. and Harvey, I. (1998). Recombination and error thresholds in finite populations. In Banzhaf, W. and Reeves, C., editors, *Foundations of Genetic Algorithms (FOGA-5)*, San Francisco, CA. Morgan Kauffman.

[Schaffer et al., 1989] Schaffer, J., Caruana, R., Eshelman, L., and Das, R. (1989). A study of control parameters affecting online performance of genetic algorithms for function optimization. In Schaffer, J. D., editor, *Proceedings of the 3rd ICGA*, San Mateo CA. Morgan Kaufmann.

[Spears, 1998] Spears, W. M. (1998). *The Role of Mutation and Recombination in Evolutionary Algorithms*. PhD thesis, George Mason University, Fairfax, Virginia.