

Assortative Mating in Genetic Algorithms for Dynamic Problems

Gabriela Ochoa, Christian Madler-Kron, Ricardo Rodriguez, and Klaus Jaffe

Universidad Simon Bolivar, Caracas 89000, Venezuela
gabro@ldc.usb.ve, christian@sauron.sytes.net

Abstract. Non-random mating seems to be the norm in nature among sexual organisms. A common mating criteria among animals is assortative mating, where individuals mate according to their phenotype similarities (or dissimilarities). This paper explores the effect of including assortative mating in genetic algorithms for dynamic problems. A wide range of mutation rates was explored, since comparative results were found to change drastically for different mutation rates. The strategy for selecting mates was found to interact with the mutation rate value: low mutation rates were the best choice for dissortative mating, medium mutation values for the standard GA, and higher mutation rates for assortative mating. Thus, GA efficiency is related to mate selection strategies in connection with mutation values. For low mutation rates typically used in GA, dissortative mating was shown to be a robust and promising strategy for dynamic problems.

1 Introduction

A source of inspiration toward understanding and improving the application of GAs is still natural evolution. Biological evolution is not possible without reproduction, and among sexual organisms, reproduction is not possible without mating or the fusion of two gametes. Mating is very unlikely to be random in nature, and mate selection may be as important in guiding evolution than natural selection. Theoretical studies of mate selection using agent-based simulations [8, 9, 6], suggest that some mating strategies confer higher fitness to individuals, and produce higher population diversity than random mating.

Most previous work on non-random mating in GAs, refers to incest-prevention techniques, where the idea is to prevent recombination between related individuals [2, 1, 3]. Other authors [5, 4] explore the inclusion of assortative mating within GAs. Assortative mating is a form of *non-random mating* common in nature, where individuals of similar phenotype mate more or less often than expected by chance. It is positive if similar organism mate more often, and negative (or dissortative) if dissimilar organisms mate with higher frequencies. All previous research discussed above deal with stationary problems, and in general report that both incest prevention techniques and negative assortative mating maintain a higher population diversity, which in turn allows a broader exploration of the

search space, and may speed up the searching process for the global optimum. Since it is believed that population diversity is an important issue for non-stationary problems, it seems worth exploring the effect of non-random mating in this context. The mutation rate has also implications on population diversity, and results can turn upside down with different mutation values. Therefore, in this work, several mutation values were explored. Moreover, both haploid and diploid representations were considered.

2 Methods

All experiments were run using a generational GA with tournament selection (tournament size of 2) and a population of 100 individuals. The genetic operations were 2-point crossover with a rate of 0.8, and the standard bit mutation. Mutation rates were expressed as mutations per genotype; several mutation values were tested (ranging from 0.0 to up to 6.0 mutations per genotype, with a step of 0.5). Both haploid and diploid representation were considered and the GA was run in three modes: (i) using both mutation and recombination in the standard way (**GA**), (ii) implementing dissortative mating (**GA-Dsrt**), and (iii) implementing assortative mating (**GA-Asrt**). Assortative mating was implemented as follows: when selecting two individuals for a crossover, the first parent was selected as usual. To chose the second parent, a set of p (*pool size*) individuals were selected using the GA fitness-based selection method. Thereafter, the similarity between each of these p genotypes and the first parent was computed. For dissortative mating, the genotype with less similarity was chosen. For assortative mating, the genotype closer to the first parent was selected as the second parent. For the experiments reported here, Hamming distance was used as the similarity measure, and the pool size p was set to 3.

2.1 Haploid/Diploid Encoding

We implemented both haploid and diploid genotypes. For the haploid encoding, a single chromosome of length n directly represents the individual's phenotype. The diploid scheme consists of a genome made up by two chromosomes of length n , and a phenotype which is obtained by applying a dominance map between each genotypic allele. The dominance map used, called *additive dominance* was originally proposed by Ryan [10]. Instead of binary digits, each locus in the genotype can be assigned a numeric value in the set $\{2, 3, 7, 9\}$. If the sum of the two alleles in a given locus is more than 10, then the phenotypic expression of this gene is 1, otherwise it is 0. In order to obtain better results when using the diploid encoding, we decided to extend Ryan's additive map with the dominance change mechanism suggested by Lewis et al. [7].

2.2 Test Problem

We used an oscillating version of the traditional single knapsack problem. The objective is to fill a knapsack with the greatest number of objects from a set of

size n , such that the total weight of the included objects is as close as possible to a target t . This implementation is based on the test problem suggested by Lewis et al. [7]. A solution is represented by a binary string of length n . This string represents the phenotype of an individual, where each gene x_i can be 1 or 0, indicating whether the object is included in the knapsack or not. The fitness f of a solution \bar{x} is given by

$$f(\bar{x}) = \frac{1}{1 + |\text{target} - \sum_{i=1}^n w_i x_i|} \quad (1)$$

For the experiments in this article, 28 objects were used. Each object had a weight $w_i = 2^i$ where i ranged from 0 to 27. This guarantees that each target is reachable by a unique combination of objects in the knapsack. This is a normalized function, values for $f(\bar{x})$ lie between 0 and 1.

Target Changes. We considered both oscillating fixed targets and randomly generated targets. When using fixed targets, every s generations the algorithm switched between two fixed targets. These targets, t_1 and t_2 , are initially randomly selected with a Hamming distance of 5. When considering random targets, the algorithm switches every s generations between newly created random targets with no specific Hamming distance. We tested two switching periods s : of 50 and 100 generations. For each experiment 10 cycles were considered, producing runs of 500 and 1000 generations respectively.

2.3 Performance Measures and Plots

For estimating optimal mutation rates in GAs we need to define what an optimal or near-optimal mutation rate is. The working definition used here is: an optimal mutation rate is the one that produces optimal performance. But then, we need a good way of measuring GA performance. We selected offline performance, that is, the mean of the current best fitness values through the whole run. Averages of 50 runs were considered. These values were plotted for several mutation rates and each reproductive strategy. We will call these plots the *offline-performance* plots.

In order to have a dynamic view of the different strategies' performance, the best fitness (averaged over 50 runs) was plotted for each generation. In these plots, the optimal (or near optimal) mutation rate, as observed from the offline performance plots, was selected for each reproductive strategy. From now onwards we will refer to these plots as: *best-fitness-trace* plots.

3 Results

Figure 1 shows results for the oscillating knapsack with two fixed targets, and a switching period of 100 generations. Ten oscillating cycles, that is, 1000 generations were considered.

From the offline-performance plots (Figure 1 (a) and (c)). We can distinguish 3 different ranges of the mutation rate; for each of these ranges, one of the mating

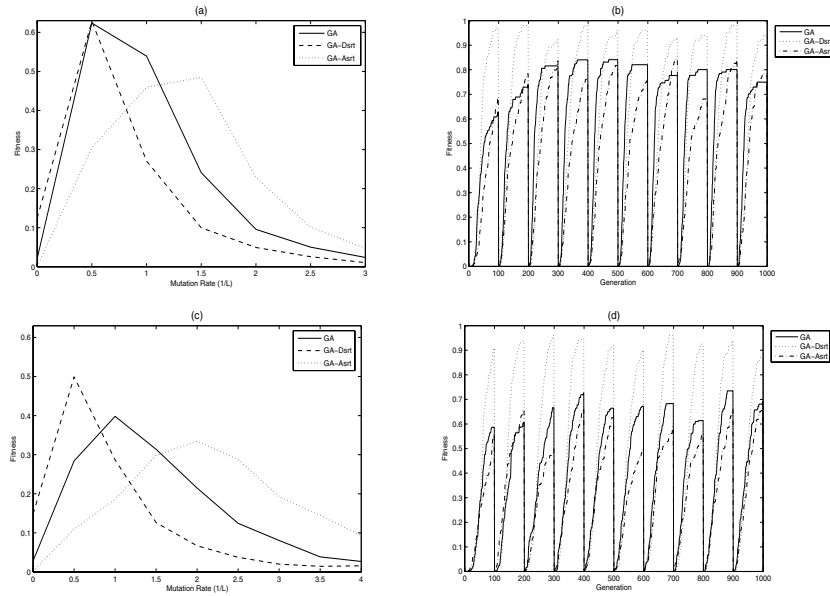


Fig. 1. Performance for fixed oscillating targets with a switching period of 100 generations. (a) Offline-performance for the haploid encoding, (b) Best-performance-trace for the haploid encoding, (c) Offline-performance for the diploid encoding. (d) Best-performance-trace for the diploid encoding. For the trace plots, the optimal mutation rate for each strategy was selected

strategies achieves the higher performance: Low mutation rates represent an advantage for GA-Dsrt, classical GA outperforms other methods at medium mutation rates, while high mutation rates create better conditions for GA-Asrt. Similar results were also found for a shorter switching period of 50 generations.

Notice from the best-fitness-trace plots (Figure 1 (b) and (d)), that the best performance (highest peaks) are delivered by GA-Dsrt in combination with a mutation rate of 0.5. Fitness values obtained with haploid individuals are slightly better than those obtained with diploid individuals. Again, similar results were found for a switching period of 50 generations.

Although the offline-performance plots show sometimes similar performance for the GA strategy as compared to GA-Dsrt (Figure 1 (a)), this was not the case when looking in detail at the corresponding generational best-fitness-trace plot: the GA strategy reaches good results, but its peaks are nonetheless almost always occluded by GA-Dsrt, which in each experiment reaches the highest peaks. This contrast was caused by the different convergence velocities of the rapidly evolving GA strategy vs. the somewhat slower dissortative strategy. Both strategies have their pros and cons: the more traditional GA may have faster convergence, but GA-Dsrt reaches higher peaks due to its inherent capacity to maintain a higher population diversity.

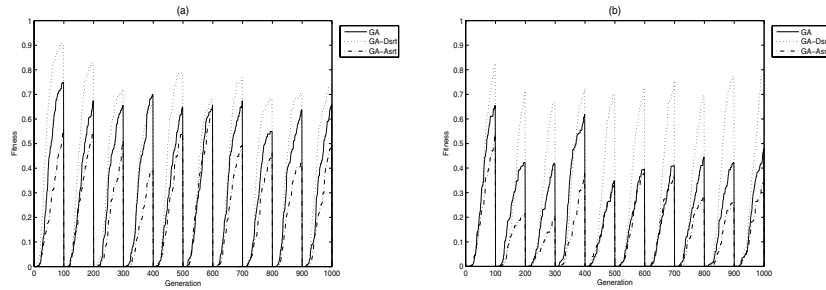


Fig. 2. Best-performance-trace for random oscillating targets with a switching period of 100 generations. (a)haploid encoding, (b) diploid encoding. The optimal mutation rate for each strategy was selected

Figures 2 show the best-performance-trace plots with random changing targets, for a switching period of 100 generations. Again, dissortative mating for both haploid and diploid individuals, is the strategy that achieves the highest fitness values (highest peaks) on each cycle. Similar results were found for the shorter switching period of 50 generations, with the exception of diploid organisms, where performance was very poor since the population had not enough time to adapt.

4 Discussion

Results suggest that the mutation rate parameter interacts with the mating strategy. Optimal mutation rates are different for each strategy; thus, fair comparisons can not be performed without selecting the optimal mutation rate for each case. When considering offline performance, on all explored scenarios dissortative mating consistently required a lower mutation rate (close to $0.5/L$) to perform better, whilst the standard GA needed a somewhat higher mutation rate (between 0.5 to 1.0) mutations per genotype) to produce similar performance. On the other hand, assortative mating produced the worst performance, and its optimal mutation rate was in the range of 1.5 to 2.5 mutations per genotype. If the optimal mutation rate was selected for each strategy, dissortative mating was found to be a good strategy for finding the highest peaks.

Surprisingly, in our experiments the haploid scheme produced better results than the diploid encoding. These results confirm the observations by Lewis et al. [7] that diploid schemes do not seem to be a robust mechanism for non-stationary problems. According to them, adding some form of dominance change considerably improves matters, but the form of the change mechanism can have a significant effect.

Mate selection is a force guiding natural evolution that has not been widely explored within artificial evolution. It is clear that selecting mates have an impact on both the search process and the population diversity. This work is a

preliminary assessment of the effect of including mate selection in evolutionary algorithms for dynamic problems. Dissortative mating was shown to be a promising scheme that may improve the algorithm performance. Further work with other non-stationary problems will make it possible to assess if these results can be generalised.

Acknowledgments: The first author thanks the Santa Fe Complex Systems Summer School organizers for providing such a nice and stimulating school. The preliminary experiments and ideas of this paper were developed in the context of SFI CSSS 2002.

References

- [1] Agostinho C. da Rosa Carlos Fernandes, Rui Tavares, *niGAVaPS - outbreeding in genetic algorithms*, Proceedings of the 2000 ACM Symposium on Applied Computing (Villa Olmo, Como, Italy), 2000.
- [2] Rob Craighurst and Worthy Martin, *Enhancing GA performance through crossover prohibitions based on ancestry*, Proceedings of the 6th International Conference on Genetic Algorithms (San Francisco) (Larry J. Eshelman, ed.), Morgan Kaufmann Publishers, July 15–19 1995, pp. 130–135.
- [3] Larry J. Eshelman and J. David Schaffer, *Preventing premature convergence in genetic algorithms by preventing incest*, Proceedings of the 4th International Conference on Genetic Algorithms (San Diego, CA) (Lashon B. Belew, Richard K.; Booker, ed.), Morgan Kaufmann, 1991, pp. 115–122.
- [4] Carlos Fernandes, Rui Taveres, Cristian Munteanu, and Agostinho C. Rosa, *Using assortative mating in genetic algorithms for vector quantization problems*, Selected Areas in Cryptography, 2001, pp. 361–365.
- [5] C.-F. Huang, *An analysis of mate selection in genetic algorithms*, Tech. report, Center for the Study of Complex Systems, University of Michigan, 2001.
- [6] Klaus Jaffe, *On sex, mate selection and evolution: an exploration*, Comments on Theoretical Biology **7** (2002), no. 2, 91–107.
- [7] Jonathan Lewis, Emma Hart, and Graeme Ritchie, *A comparison of dominance mechanisms and simple mutation on non-stationary problems*, Parallel Problem Solving from Nature – PPSN V (Berlin) (Agoston E. Eiben, Thomas Bäck, Marc Schoenauer, and Hans-Paul Schwefel, eds.), Springer, 1998, Lecture Notes in Computer Science 1498, pp. 139–148.
- [8] G. F. Miller and P. M. Todd, *The role of mate choice in biocomputation: Sexual selection as a process of search, optimization and diversification*, Lecture Notes in Computer Science **899** (1995), 169–198.
- [9] Gabriela Ochoa and Klaus Jaffe, *On sex, mate selection and the red queen*, Journal of Theoretical Biology **199** (1999), 1–9.
- [10] Conor Ryan, *The degree of oneness*, 1st Online Workshop on Soft Computing (Nagoya, Japan), Nagoya University, 1996, pp. 100–105.