Emergent gender differentiation in gender genetic algorithm meta-optimization[1]

Christopher Briggs
Department of Mathematics
Embry-Riddle Aeronautical University
Prescott, AZ USA
briggsc1@erau.edu

Abstract

Genetic algorithms are metaheuristics first presented in their modern form by Holland in 1992. There are many variations, but all involve an evolving population of candidate solutions to some problem. While most implementations involve asexual reproduction, some attempts have been made to harness advantages of sexual reproduction in genetic algorithms - such approaches are called gender genetic algorithms. In particular, the male mutation bias, which has been well documented in mammals, depends on gender differentiation in parameters relevant to genetic algorithms. We implement gender and perform meta-optimization on the onemax benchmark and statistically analyze whether gender differences in mutation rate and tournament size naturally emerge.

1 Introduction

Metaheuristic algorithms attempt to find good solutions to optimization problems. Metaheuristic algorithms may be divided into solution-based and population-based methods. Solution-based methods, such as simulated annealing and tabu search, operate on a single candidate solution. These methods may be susceptible to fixing near local optima. Population-based methods offer an alternative approach: multiple candidate solutions develop in tandem. Some examples of population-based methods are particle swarm, ant colony, and genetic algorithms (GA) [8]. Metaheuristics, GA included, offer practical ways to find good solutions in reasonable time to problems with intractably large brute-force search spaces, for example the 0–1 knapsack problem [11].

The idea of applying evolutionary principles in a computing environment was arguably first conceived by Alan Turing in 1950. He wrote of an education process for programs and noted the connection with the evolutionary

[1]A similar summary for a similar presentation previously appeared in the 2022 Proceedings of the Exchange of Mathematical Ideas.

process [13]. In 1954 at Princeton's Institute for Advanced Study, Barricelli first used a computer to simulate evolution, including genetic code [1]. Genetic algorithms in their modern form - as metaheuristics - were introduced by Holland in 1992 [6]. There are many variations on GA, but the essential core is: a population of candidate solutions, represented by genomes and judged by a "fitness" function, evolve and reproduce according to Darwinian and Mendelian principles respectively.

In addition to a function which evaluates the fitness of a genome, GA standardly include the following components: initialization, in which the population is initialized; selection, in which population members are selected for reproduction; crossover, in which population members reproduce by combining genetic information; mutation, by which randomness is injected to the genome; and termination, or rules for conclusion of the process. There are many variations on the specifics of implementation, such as how population members are chosen for reproduction, how genomes are combined in reproduction, what types of mutations occur, and how the population is culled as new members are created. Of particular interest to the present study are the selection and mutation components.

In 1996, Lis and Eiben introduced gender as a feature in genetic algorithms, an approach which has come to be called gender genetic algorithms (GGA). Their approach was created a gender for each criterion in a multi-criteria optimization problem [9]. In the early 2000s, advances in genomic sequences allowed estimation of male and female contribution to genome mutation across species; generally the male contribution is higher than the female contribution [10]. In 2003, Sánchez-Velazco and Bullinaria incorporated this insight into a GGA model by differentiating population member mutation rates by gender. They tried several values of hyperparameters (e.g. controlling sexual selection), realizing improvements in convergence speed over genderless implementations [2]. There soon followed other examples of GGA implementations realizing performance gains over genderless implementations against various benchmarks [2, 5].

2 Methodology

Similar to Sánchez-Velazco and Bullinaria, the present study explores GGA with population members potentially differentiated by gender in selection and mutation. Rather than sampling several values for the involved hyperparameters, we use GA for meta-optimization of the GGA hyperparameters. Rather than measure efficiency and robustness of resulting GGA models versus genderless implementations, we statistically analyze evidence for emergence of

differentiated genders in the GGA model during meta-optimization.

3 Component functions

3.1 Gender genetic algorithm

For GGA, the population is initialized with 100 members of each of two genders (labeled "male" and "female"). The chromosome length is 100, and the benchmark is onemax. The tournament function randomly selects a fixed number (determined by gender) of females and a (possibly different) fixed number of males from the population, then selects the fittest of each of these two subsets. These two fittest chromosomes are recombined in the crossover function. The number of females and males selected are referred to as tournament sizes.

The crossover function propagates each parent unchanged to the next generation with probability 0.1. Otherwise, a position of the chromosome is randomly selected, and two offspring are created by appending the first part of each parent's chromosome to the second part of the other parent's chromosome. One of the offspring is designated female and the other male, and they are added to the next generation. Mutation is executed independently allele-by-allele following crossover. Each gender has a per-allele mutation probability between 0 and 0.025.

The selection, crossover, and mutation functions iterate until the next generation reaches the size of the initial population.

Termination conditions should generally depend on the benchmark. In the case of onemax, termination occurs when there has been no improvement for 100 generations or when 5000 total generations have elapsed.

3.2 Genetic algorithm meta-optimization

The purpose of the GA meta-optimization is to tune hyperparemeters of the GGA population. Specifically, the male and female tournament sizes and mutation rates are modulated. A genderless meta-population is initialized with 20 members. The chromosome length is 24. The chromosome encodes values for the tournament sizes and mutation rates hyperparameters of an execution of GGA. The first four bits of the meta-population chromosome encode the female tournament size (between 10 and 25), the next four bits encode the male tournament size (also between 10 and 25), the next eight bits encode the female mutation rate (between 0 and 0.025), and the final eight bits encode the male mutation rate (also between 0 and 0.025). The

GA tournament size is 8, with the fittest of each of two random population selections reproducing. Parents propagate unchanged to the next generation with probability 0.1. Otherwise, a position of the chromosome is randomly selected, and two offspring are created by appending the first part of each parent's chromosome to the second part of the other parent's chromosome. The offspring are mutated before being added to the next generation. Mutation is carried out independently allele-by-allele with mutation probability $\frac{1}{24}$ per allele.

The fitness function for the GA meta-optimization uses a number of executions to termination of the underlying GGA. Each execution is scored by how many generations it takes for an optimal solution to be acquired (subject to termination conditions described above). The fitness of the GA meta-population member is the average of these scores.

Take, for example, the GA population member

$$a_0, a_1, \ldots, a_{23} = 001011011000001100100100.$$

This chromosome determines the following GGA hyperparameters:

$$\begin{aligned}
\text{tourn_size_f} &= 10 + \sum_{n=0}^{3} 2^{3-n} a_n = 12, \\
\text{tourn_size_m} &= 10 + \sum_{n=4}^{7} 2^{7-n} a_n = 23, \\
\text{mutate_rate_f} &= \sum_{n=8}^{15} 2^{15-n} a_n \left(\frac{0.025}{2^8 - 1} \right) \approx 0.0128, \\
\text{mutate_rate_m} &= \sum_{n=16}^{23} 2^{23-n} a_n \left(\frac{0.025}{2^8 - 1} \right) \approx 0.00353.
\end{aligned} \tag{1}$$

The fitness of this meta-population member is found by averaging the scores of many executions of the underlying GGA using these hyperparameters.

4 Results

4.1 Onemax

Onemax is a simple optimization problem in which the optimum (in this implementation, highest) sum of a bit string is sought. Clearly, for a bit string of n bits, the optimum is n. We used $n = 100$. The fitness of a GA population member is the average number of generations required to

reach the optimal solution over 50 executions of the corresponding GGA, with 150 returned should the GGA terminate after no improvement for 100 generations. It does occur in practice that some executions of the GGA require more than 150 generations to terminate, but this does not matter as within several generations the GA selection pressure improves typical GA population member fitness to numbers well below 150.

The GA meta-optimization was run for 5000 generations, with the fittest chromosome of each generation recorded along with its measured fitness. We observe that the genders were quickly differentiated and remained differentiated for the duration of the experiment. The female and male mutation probabilities encoded by the fittest member of each GA generation are shown in Figure 1. The female and male mutation probabilities are dependent, $\chi^2(4752, N = 10000) = 9288.09,\ p < .001$. The median female mutation probability is 0.0125, while the median male mutation probability is 0.0160. The female and male tournament sizes encoded by the fittest member of each GA generation are shown in Figure 2. The median female tournament size is 20, while the median male tournament size is 25. While the distributions of male and female tournament sizes are distinct, unlike the mutation probabilities they are independent, $\chi^2(169, N = 10000) = 107.72,\ p > .1$.

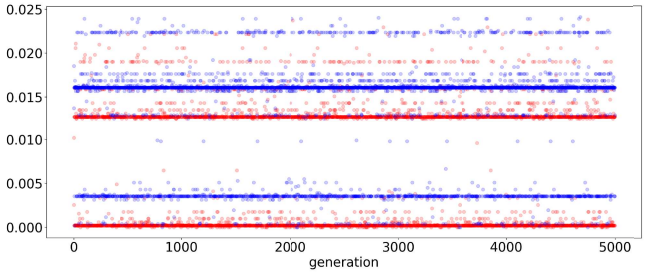

Figure 1: Mutation probabilities of each gender encoded by the fittest population member of each of 5000 GA generations. Red is female and blue is male.

While this experiment shows the male subpopulation developing higher mutation rates in agreement with biological observation, the labels of "female" and "male" are arbitrary as the population members begin undifferentiated. The gender differences emerges through selection pressure, and it would be equally likely for the "female" subpopulation to develop the higher mutation rate.

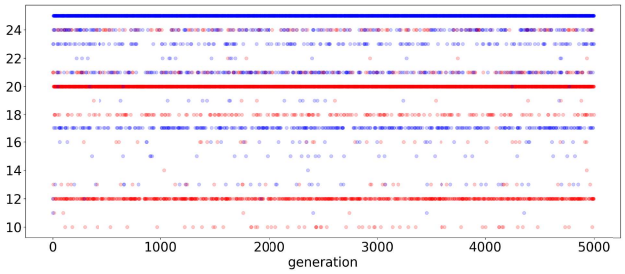

Figure 2: Tournament sizes for each gender encoded by the fittest population member of each of 5000 GA generations. Red is female and blue is male.

4.2 Lookup table failure

The validity of the above results are in question due to a probable error in the code which generated these results. Because the meta population fitness function is computationally expensive, and it must be computed very many times for a given population member, a fitness once computed was stored in a lookup table. This way, the fitness for a given population member would be computed only once. This was necessary to run the algorithm in practical time on available equipment. However, the computed fitness scores of a few population members were anomalously low; once stored in the lookup table, these genomes dominated future generations. Evidence for this phenomenon can be seen in the horizontal lines of 1 and 2.

4.3 Future work

The problem identified above will first be addressed. Once it is resolved, statistical analysis will evaluate the hypothesis that spontaneous gender differentiation occurs under GA selection pressure. Greater computing power may be utilized to extend the replicate the study in the context of other benchmark functions the location of whose optima is more computationally intensive, for example Rosenbrock's function, called F2 by De Jong [4].

5 Limitation

After collecting results, if there is evidence for the spontaneous gender differentiation, it would be tempting to speculate that the evidence offers a partial explanation for the emergence of sex among organisms. We do not so speculate, referring instead to research pointing to well-established theoret-

ical considerations of the origin, including by parasitic genetic elements [3], cannibalism [7], or vaccination [12].

References

[1] N. Barricelli, Esempi numerici di processi di evoluzione, *Methodos*, **6**, 45–68 (1954).

[2] J. Sanchez-Velazco, J. Bullinaria, Gendered selection strategies in genetic algorithms for optimization, *UKCI*, 217–223 (2003).

[3] S. DasSarma, Extreme Microbes, *American Scientist*, **95**, 224–231 (2007).

[4] K. De Jong, An Analysis of the Behavior of a Class of Genetic Adaptive Systems, *Ph.D. Thesis, University of Michigan, Ann Arbor*, pp 197-200 (1975).

[5] T. Drezner and Z. Drezner, Gender-specific genetic algorithms, *INFOR: Information Systems and Operational Research*, **44**, 117–127 (2006).

[6] J. H. Holland, Genetic algorithms, *Scientific American*, **267**, 66–73 (1992).

[7] W. Sterrer, On the origin of sex as vaccination, *Journal of Theoretical Biology*, **216**, 387–396 (2002).

[8] S. Katoch, S. S. Chauhan, and V. Kumar, A review on genetic algorithm: past, present, and future, *Multimed Tools Appl*, **80**, 8091–8126 (2021).

[9] J. Lis and A. E. Eiben, A multi-sexual genetic algorithm for multiobjective optimization, *Proceedings of 1997 IEEE International Conference on Evolutionary Computation*, 59–64 (1997).

[10] M. A. Wilson Sayres and K. D. Makova, Genome analyses substantiate male mutation bias in many species, *Bioessays*, **33**, 938–45 (2011).

[11] R. P. Singh, Solving 0–1 Knapsack problem using genetic algorithms, *2011 IEEE 3rd International Conference on Communication Software and Networks*, 591–595 (2011).

[12] O. Judson, *Dr. Tatiana's sex advice to all creation*, Metropolitan Books, New York, pp. 233–4. 2002.

[13] A. M. Turing, Computing Machinery and Intelligence, *Mind*, **59**, 433–460 (1950).