The Allele Meta-Model – Developing a Common Language for Genetic Algorithms

Stefan Wagner, Michael Affenzeller

Institute for Formal Models and Verification
Johannes Kepler University
Altenbergerstrasse 69
A-4040 Linz - Austria
{stefan,michael}@heuristiclab.com

Abstract. Due to the lot of different Genetic Algorithm variants, encodings, and attacked problems, very little general theory is available to explain the internal functioning of Genetic Algorithms. Consequently it is very difficult for researchers to find a common language to document quality improvements of newly developed algorithms.
In this paper the authors present a new Allele Meta-Model enabling a problem-independent description of the search process inside Genetic Algorithms. Based upon this meta-model new measurement values are introduced that can be used to measure genetic diversity, genetic flexibility, or optimization potential of an algorithm's population. On the one hand these values help Genetic Algorithm researchers to understand algorithms better and to illustrate newly developed techniques more clearly. On the other hand they are also meaningful for any GA user e.g. to tune parameters or to identify performance problems.

1 Introduction

Genetic Algorithms (GAs) developed in 1975 by J. H. Holland [11] are a heuristic optimization technique based on the natural evolution process. Although they represent a very strong simplification of the complex processes observed in nature we commonly subsume with the term evolution, GAs are very successful in different fields of industrial and scientific application. In spite of their success there is still very little mathematical theory available that might comprehensively explain the different processes and their interactions inside a GA. One of the main reasons for this drawback is the tremendous lot of different GA variants, encodings, operators, and attacked problems. So most theoretical considerations like the Schema Theorem introduced by J. H. Holland [11] or the Building Block Hypothesis presented by D. E. Goldberg [9] concentrate mainly on one specific form of individual encoding (in most cases binary encoding) and can therefore hardly be generalized. Other approaches aiming to develop a common theory for Evolutionary Computation in general (cf. [14], [18]) are faced with severe difficulties due to the huge variety in the field of GAs.

As a consequence of this lack of theoretical background most scientists working in the area of GAs have a very profound intuitive knowledge about GAs

and GA behavior but it is very difficult for them to show some kind of hard facts documenting their results. Most GA researchers will agree on the fact that effects like selection, selection pressure, genetic diversity, (unwanted) mutations, or premature convergence strongly interact with each other. These interactions play an important role concerning achievable solution quality. However, very few papers can be found that suggest ways to measure and visualize these forces affecting GA populations.

In this contribution the authors present some new measurement values that might help to overcome this situation. First of all a new allele oriented meta-model for GAs is presented making it possible to understand and to discuss processes inside the algorithm in a problem-independent way. Based upon this allele model new measurement values are presented measuring genetic diversity, genetic variety, goal-orientedness, and optimization potential during a GA run. So these values help any GA developer e.g. to precisely show improvements when introducing new algorithm variants and to discuss and compare the behavior of new algorithms with already existing ones. Furthermore, the values can also be used by any GA user to get some more meaningful feedback from the algorithm helping to tune parameters, to identify performance problems, or to develop a deeper understanding for GAs in general.

2 An Allele Oriented Meta-Model for GAs

GAs mainly use the three genetic operators selection, crossover, and mutation to manipulate solution candidates in order to achieve better results. Only one of these three operators, namely selection, is independent of the chosen problem encoding, as selection depends only on the individuals' fitness values. So when trying to make any general propositions about GAs this high amount of problem (i.e. encoding) dependency is a severe difficulty. Therefore, most of the existing theoretical considerations like the Schema Theorem or the Building Block Hypothesis only concentrate on binary encoding, as it was the first encoding variant suggested by J. H. Holland and is still used in numerous applications.

However, in the last years various other forms of individual encoding (e.g. permutation-based, real-valued, etc.) have been introduced. These codifications have shown high potential in lots of applications like e.g. the Traveling Salesman Problem, Scheduling Problems, or Genetic Programming. In those cases using binary encoding was not at all intuitive and required the development of sophisticated and rather inappropriate crossover and mutation operators. As different solution encodings also led to the development of new crossover and mutation operators, it is very difficult to generalize the theoretical statements of Holland and Goldberg in order to be applicable for these new GA variants.

So before we can think of new measurement values describing the internal functioning of GAs, it is necessary to develop a new problem-independent view. Otherwise the newly proposed insights would also only be applicable for a specific kind of GA applications. This conclusion was the cornerstone for the development of the *Allele Meta-Model* of GAs. The basic question that needs to be answered

is, what the atomic entities, GAs work with, are and if and how these entities can be stated in a problem-independent way. In fact, in the case of binary encoding the Building Block Hypothesis already highlighted the importance of small parts of genetic material (low-order schemata with short defining length and above average fitness) that are assembled by the algorithm to generate better solutions. When abstracting this consideration a GA in general can be seen as a process that combines different parts of genetic code.

In biology the concrete realization of a gene is called an allele and represents the logical entity on top of the molecular level. So it seems quite reasonable to use the term allele also in the context of GAs describing the basic entity that represents genetic information, forms chromosomes, and describes traits. In the case of binary encoding an allele can e.g. be a bit at a specific position of the individual's bit string. As the concept of alleles is problem-independent and not restricted to binary encoding, it can be defined for other GA encodings as well. In the case of permutation encoding e.g. the sequence of two numbers of the permutation or a number at a specific position of the permutation can be considered as an allele depending on the optimization problem and the interpretation of its encoding. Obviously the identification and interpretation of alleles, i.e. the layer below the Allele Meta-Model, is highly problem and encoding dependent. Though on top of the Allele Meta-Model GAs can be discussed in a completely problem-independent way including crossover and mutation concepts.

Based upon the Allele Meta-Model the Standard Genetic Algorithm (SGA) (as described in e.g. [7], [8], [16], [17], [21]) can be reformulated in the following way: Each individual is represented by a randomly initialized set of alleles. Each set is analyzed by an evaluation function returning a fitness value of the individual. In the selection step individuals, i.e. allele sets, with high fitness are selected for reproduction. Then the crossover operator is used to merge two parental allele sets to one new set in order to generate a new solution, i.e. child. Thereby it might happen, that not all alleles contained in the new solution are also elements of at least one of the parental allele sets. This situation occurs when using some more complex encoding requiring crossover operators with some kind of repair strategy and is referred to as unwanted mutations. Finally the mutation operator might be used to exchange some alleles of the child's allele set by some other ones.

3 Allele Frequency Analysis

As population genetics focus on the changing of allele frequencies in natural populations the Allele Meta-Model of GAs builds the bridge between GAs and population genetics. So it should be a very fruitful approach to consider various aspects and terms of population genetics for GA analysis (cf. [2], [3]). Population genetics define various forces influencing allele frequencies in natural populations: the Hardy-Weinberg Law, Genetic Drift, Selection, Mutation, etc. (a good overview can be found in [10]). However, as the basic population model in population genetics assumes diploid individuals these insights have to be adapted

accordingly in order to be valid for GAs which are haploid per design. All these different forces lead to one of the following four different results concerning the frequency of alleles (p denotes the probability that a specific allele is contained in the genetic code of an individual, or in other words is element of an individual's allele set):

- $\mathbf{p} \rightarrow \mathbf{1}$: The allele is fixed in the entire population.
- $\mathbf{p} \rightarrow \mathbf{0}$: The allele is lost in the entire population.
- $\mathbf{p} \rightarrow \hat{\mathbf{p}}$: The allele frequency converges to an equilibrium state.
- $\mathbf{p} \rightarrow \mathbf{p}$: The allele frequency remains unchanged.

So in this context the global goal of GAs can be reformulated in the following way: Use selection, crossover and mutation to modify the allele frequencies of the population in such a way that all alleles of a global optimal solution are fixed in the entire population. All other alleles belonging to suboptimal solutions should be lost.

As a consequence it should be very insightful to monitor the distribution of alleles in a GA population during the execution of the algorithm in order to observe the success of a GA concerning the success criterion stated above. In fact this is the main idea of the Allele Frequency Analysis (AFA) and its new measurement values. Consequently it is necessary to distinguish between two different types of alleles: On the one hand there are alleles belonging to a global optimal solution and on the other hand all other alleles being definitively not optimal. As a matter of course such a distinction can only be made when using benchmark problems with known optimal solutions. For better differentiation the first kind of alleles are referred to as *relevant alleles* (cf. building blocks) in the following.

Based on the Allele Meta-Model of GAs some measurement values can be introduced that represent the basis of the AFA:

- **Total Number of Different Alleles (A):**
  The total number of different alleles contained in the entire population is a precise measurement for genetic diversity. The more different alleles are available in the population the more diverse is the genetic code of the individuals. In the case of combinatorial optimization problems it's important to bear in mind that the total number of different alleles in the whole solution space is usually not that large as the complexity of combinatorial optimization problems is caused by the millions of possible combinations of alleles. E.g. a 100 cities Traveling Salesman Problem has only $99 + 98 + 97 + \ldots + 1 = \frac{100 \cdot 99}{2} = 4950$ different edges, i.e. alleles.
- **Number of Fixed Alleles (FA):**
  A second measurement value of interest is the number of fixed alleles in the entire population, i.e. the number of alleles contained in the allele set of every individual. It indicates the genetic flexibility of the population as any fixed allele cannot be altered by crossover anymore (apart from unwanted mutations). Consequently especially the fixing of suboptimal alleles is very harmful for GA performance because it benefits premature convergence.

– **Total Number of Relevant Alleles (RA):**
If benchmark problems with known optimal solutions are used for analyzing GAs, it is possible to identify alleles of global optimal solutions and to count their total number in the entire population. This value provides information about how goal-oriented the evolutionary search process is. Ideally the total number of relevant alleles should be steadily increasing until all relevant alleles are fixed in the entire population.

– **Number of Fixed Relevant Alleles (FRA):**
The number of fixed relevant alleles is a fraction of all fixed alleles and estimates the success chances of the GA. Especially the deviation between the number of fixed relevant and fixed alleles is very informative as it points out how many suboptimal alleles are already fixed in the population which indicates the severity of premature convergence.

– **Number of Lost Relevant Alleles (LRA):**
Contrary to the number of fixed relevant alleles the number of lost relevant alleles shows how many relevant alleles are not included in the population's gene pool anymore. In an ideal (hyperplane sampling) GA this value should always be 0 as such lost relevant alleles can only be regained by mutation which is contradictory to the idea of hyperplane sampling. Again this measurement value helps to appraise premature convergence.

– **Distribution of Relevant Alleles (DRA):**
Concerning relevant alleles there is finally the opportunity to monitor the distribution of all relevant alleles in the entire population during the GA execution. In fact this is not a single measurement value but a very good technique to visualize the dynamics of a GA and in particular the interplay between hyperplane sampling (crossover) and neighborhood search (mutation).

– **Selection Pressure (SP):**
Last but not least there is another measurement value not directly motivated by the Allele Meta-Model. The concept of selection pressure was first introduced by Charles Darwin as a result of birth surplus [6]: A population is producing more offspring than the actual environmental resources can keep alive. Consequently some of the not so fit children die before they reach the age of sexual maturity. This fact causes a so-called selection pressure among the offspring requiring a minimum fitness to survive in order to pass on their own genetic information. In the context of Evolutionary Computation selection pressure has been defined for some algorithms that also produce a birth surplus like Evolution Strategies (ES) [19], the Breeder GA [15], SEGA [1], or SASEGASA [4]. E.g. selection pressure is defined as $\frac{\lambda}{\mu}$ for the (μ, λ)-ES where μ denotes the population size and λ stands for the number of procreated offspring. A large value of $\frac{\lambda}{\mu}$ indicates a high selection pressure (small population size, lots of children) and vice versa. In the above mentioned algorithms selection pressure turned out to have a great influence on the algorithms' performance.

However, in the general case of GAs selection pressure cannot be defined so easily as a GA is normally procreating exactly as many children as needed. So how can selection pressure be measured, if there is no birth surplus? One possible suggestion is to define selection pressure as the ratio of the selection probability of the fittest individual to the average selection probability of all individuals (see e.g. [5]). However, this definition has two weaknesses: First, it is an a priori definition of selection pressure, which doesn't take stochastic effects into account that might have a great influence especially when using rather small populations. Second, the average selection probability depends on the chosen selection strategy and cannot be calculated that easily in some cases (e.g. when using Tournament Selection concepts). So the authors decided to calculate selection pressure in an a posteriori way independent of the used selection operator.
Selection pressure can be abstracted somehow as a measurement value indicating how hard it is for an individual to pass on its genetic characteristics from one generation to the next. So it seems to be reasonable to define selection pressure in a classical GA as the ratio between the population size and the number of individuals selected as parents of the next generation. If the individuals of the next generation are procreated by a few parent individuals only, selection pressure is very high and vice versa. So selection pressure can be calculated according to the following formula:

$$SP = 1 - \frac{|PAR|}{|POP|} \tag{1}$$

where $|PAR|$ stands for the number of different selected parents and $|POP|$ represents the population size. So a minimum selection pressure of 0 indicates that all individuals of the parental generation got the chance of mating and a maximum selection pressure of $1 - \frac{1}{|POP|}$ represents the situation that all offspring are mutated clones of a single super-individual.

4 Allele Frequency Analysis in Practice

In this section an example run of the Standard Genetic Algorithm (SGA) is performed with the HeuristicLab[1] optimization environment [22] to outline the potential of the AFA. To highlight the encoding independency of the AFA a test problem not restricted to binary encoding was chosen. The authors decided to use the Traveling Salesman Problem (TSP) (e.g. described in [13]) as the TSP is a very well-known combinatorial optimization problem with exactly one optimal solution in a majority of cases. Furthermore, a lot of different encodings, crossover operators and mutation concepts for GAs are available (cf. [12]). As test problem instance the ch130 TSP benchmark problem taken from TSPLIB [20], a comprehensive collection of TSP benchmark problems, is used. For this problem not only the quality, i.e. the tour length, of the optimal solution is known (6'110)

[1] More details can be found on the HeuristicLab homepage `http://www.heuristiclab.com`.

but also the optimal tour itself, which makes it well-suited for the AFA. Furthermore, path representation is used for solution encoding, whereby the alleles are represented by the edge information contained within the permutation in that case.

However, it has to be mentioned once again that the AFA is not restricted to a specific form of TSP encoding or to the TSP in general. The AFA can be performed for any kind of optimization problem, if the definition of alleles is reasonable.

The parameter values used for the example run are listed in Table 1. Moreover, the achieved results concerning solution quality and the AFA values of the last generation are presented in Table 2.

Table 1. Parameter Settings

Generations	2'500	Selection Operator	Tournament Selection
Population Size	250	Crossover Operator	Order Crossover (OX)
Mutation Rate	5%	Mutation Operator	Simple Inversion Mutation (SIM)
Tourn. Group Size	3		

Table 2. Solution Quality and AFA Results

Optimal Solution Fitness	6'110	A	150
Best Found Solution Fitness	7'099	RA	19'738
Average Solution Fitness	7'118.06	FA	113
Evaluated Solutions	625'000	FRA	69
Average SP	0.361	LRA	51

However, the development of solution quality and AFA values during the GA run is far more interesting than the final results as it helps to gain some insight about what's going on inside the algorithm. In Figure 1 the fitness value of the best found solution and the average solution fitness, as well as the A and RA value (second y-axis) are shown. Additionally, Figure 2 shows the progress of A as well as FA, FRA, and LRA (second y-axis). It can be seen that the diversity of the population (A) is decreasing proportional with the decrease of the fitness value.[2] Contrariwise the number of relevant alleles (RA) is increasing as the solution quality increases. These aspects are not really surprising as the GA uses evolutionary forces to improve the genetic material of its population. Consequently, disadvantageous alleles will be eliminated due to selection. The population size is not modified during the run and so the total number of alleles

[2] Note that in the context of the TSP a decreasing fitness value is equivalent to an increasing solution quality as the total tour length is used as fitness value which should be minimized.

is staying constant leading to an increase of advantageous alleles reflected in the increasing RA.

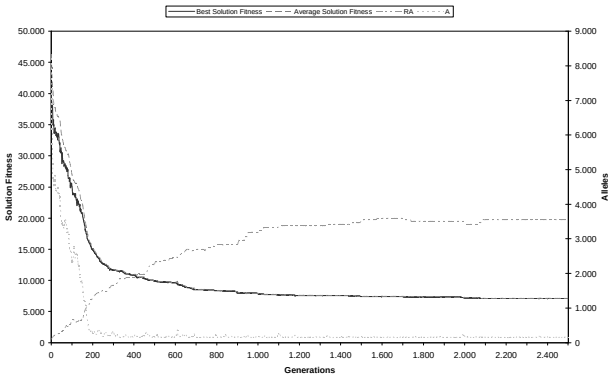

Fig. 1. Best Solution Fitness, Average Solution Fitness, A, and RA

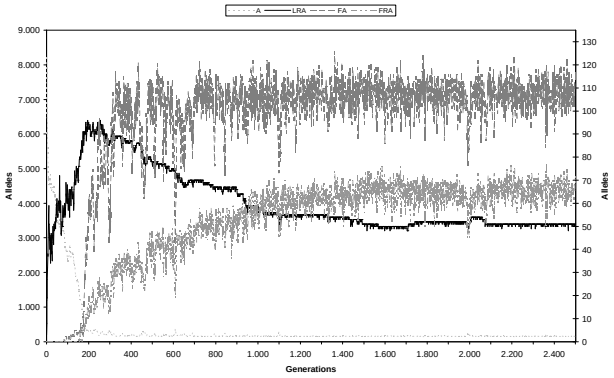

Fig. 2. A, FA, FRA, and LRA

A more interesting aspect revealed by the charts is the drastic loss of genetic diversity in the first few generations. The A value is dropping from almost 8.000 at the beginning to approximately 4.500 within the first 20 generations. This dramatic diversity reduction comes along with a significant increase of solution quality. However, also a lot of relevant alleles are lost indicated by the LRA value jumping to almost 60 during this period. Although mutation and also unwanted mutations are able to regain some of the lost relevant alleles the algorithm doesn't fully recover from this initial diversity loss during the whole run, leading to premature convergence in the end.

After this initial phase genetic diversity is further strongly decreased and on the opposite the number of lost relevant alleles increases, indicating a rather high selection pressure (as expected when using Tournament Selection with a group size of 3). Then shortly before generation 200 is reached genetic diversity is reduced that much that first alleles are fixed in the entire population (FA). The FA value increases very quickly and from that moment on genetic flexibility of the population is very low. Crossover is not able to combine alleles in order to generate better solutions anymore and the algorithm needs mutation and also unwanted mutations to induce new alleles into the gene pool of its population. Obviously, mutation is able to find some of the missing relevant alleles in the last phase of the GA as the number of relevant alleles and consequently also the solution quality further increases slowly. These newly found relevant alleles are then propagated via crossover among the allele sets of all individuals leading to an increasing FRA value. However, not all relevant alleles are regained by mutation and so the algorithm prematurely converges at a best found solution fitness value of 7'099 which is 16.19% worse than the optimal solution.

5 Conclusion

In this paper the authors present the Allele Meta-Model for GAs. By introducing alleles as the atomic entities GAs work with, it gets possible to consider the whole process of a GA in a problem-independent way. Furthermore, inspired by the area of population genetics the Allele Meta-Model builds the basis for introducing some new measurement values subsumed with the term Allele Frequency Analysis. These values describe the internal state inside an algorithm by measuring genetic diversity, genetic flexibility, goal-orientedness, or optimization potential. Furthermore, in a short experimental part the paper also illustrates how the measurement values also help to predict premature convergence and to identify its reasons.

Finally it can be stated, that the Allele Meta-Model and the Allele Frequency Analysis are not only meaningful for any GA researcher helping to document improvements of newly developed algorithms and providing a common language, but also for GA users, as the calculated values provide essential feedback about the algorithm and help to tune parameters, to identify performance problems, and to gain deeper understanding for GAs in general.

References

1. M. Affenzeller. Segregative genetic algorithms (SEGA): A hybrid superstructure upwards compatible to genetic algorithms for retarding premature convergence. *International Journal of Computers, Systems and Signals (IJCSS)*, 2(1):18–32, 2001.
2. M. Affenzeller. Population genetics and evolutionary computation: Theoretical and practical aspects. *Accepted to be published in: Journal of Systems Analysis Modelling Simulation*, 2004.

3. M. Affenzeller and S. Wagner. The influence of population genetics for the redesign of genetic algorithms. In Z. Bubnicki and A. Grzech, editors, *Proceedings of the 15^{th} International Conference on Systems Science*, volume 3, pages 53–60. Oficyna Wydawnicza Politechniki Wroclawskiej, 2004.
4. M. Affenzeller and S. Wagner. SASEGASA: A new generic parallel evolutionary algorithm for achieving highest quality results. *Journal of Heuristics - Special Issue on New Advances on Parallel Meta-Heuristics for Complex Problems*, 10:239–263, 2004.
5. T. Bäck. Selective pressure in evolutionary algorithms: A characterization of selection mechanisms. In *Proceedings of the First IEEE Conference on Evolutionary Computation*, pages 57–62. IEEE Press, 1994.
6. C. Darwin. *The Origin of Species.* Wordsworth Classics of World Literature. Wordsworth Editions Limited, 1998.
7. L. Davis. *Handbook of Genetic Algorithms.* Van Nostrand Reinhold, 1991.
8. D. Dumitrescu, B. Lazzerini, L. C. Jain, and A. Dumitrescu. *Evolutionary Computation.* The CRC Press International Series on Computational Intelligence. CRC Press, 2000.
9. D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning.* Addison Wesley Longman, 1989.
10. D. L. Hartl and A. G. Clark. *Principles of Population Genetics.* Sinauer Associates Inc., 2^{nd} edition, 1989.
11. J. H. Holland. *Adaption in Natural and Artifical Systems.* University of Michigan Press, 1975.
12. P. Larranaga, C. M. H. Kuijpers, R. H. Murga, I. Inza, and D. Dizdarevic. Genetic algorithms for the travelling salesman problem: A review of representations and operators. *Artificial Intelligence Review*, 13:129–170, 1999.
13. E. L. Lawler, J. K. Lenstra, A. Rinnooy-Kan, and D. B. Shmoys. *The Travelling Salesman Problem.* Wiley, 1985.
14. H. Mühlenbein. Towards a theory of organisms and evolving automata. In A. Menon, editor, *Frontiers of Evolutionary Computation*, volume 11 of *Genetic Algorithms and Evolutionary Computation*, chapter 1. Kluwer Academic Publishers, 2004.
15. H. Mühlenbein and D. Schlierkamp-Voosen. The science of breeding and its application to the breeder genetic algorithm BGA. *Evolutionary Computation*, 1(4):335–360, 1993.
16. Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs.* Springer-Verlag, 1992.
17. M. Mitchell. *An Introduction to Genetic Algorithms.* The MIT Press, 1996.
18. N. J. Radcliffe. Forma analysis and random respectful recombination. In R. K. Belew and L. B. Booker, editors, *Proceedings of the 4^{th} International Conference on Genetic Algorithms*, pages 222–229. Morgan Kaufmann Publishers, 1991.
19. I. Rechenberg. *Evolutionsstrategie.* Friedrich Frommann Verlag, 1973.
20. G. Reinelt. TSPLIB - A traveling salesman problem library. *ORSA Journal on Computing*, 3:376–384, 1991.
21. M. Tomassini. A survey of genetic algorithms. *Annual Reviews of Computational Physics*, 3:87–118, 1995.
22. S. Wagner and M. Affenzeller. Heuristiclab: A generic and extensible optimization environment. In *Accepted to be published in: Proceedings of ICANNGA 2005*, 2005.