A Self-Adaptive Model for Selective Pressure Handling within the Theory of Genetic Algorithms

Michael Affenzeller, Stefan Wagner

Institute of Systems Science
Systems Theory and Information Technology
Johannes Kepler University
Altenbergerstrasse 69
A-4040 Linz - Austria
{ma,sw}@cast.uni-linz.ac.at

Abstract. In this paper we introduce a new generic selection method for Genetic Algorithms. The main difference of this selection principle in contrast to conventional selection models is given by the fact that it considers not only the fitness of an individual compared to the fitness of the total population in order to determine the possibility of being selected. Additionally, in a second selection step, the fitness of an offspring is compared to the fitness of its own parents. By this means the evolutionary process is continued mainly with offspring that have been created by advantageous combination of their parents' attributes. A self-adaptive feature of this approach is realized in that way that it depends on the actual stadium of the evolutionary process how many individuals have to be created in order to produce a sufficient amount of 'successful' offspring. The experimental part of the paper documents the ability of this new selection operator to drastically improve the solution quality. Especially the bad properties of rather disadvantageous crossover operators can be compensated almost completely.

1 Introduction

Genetic Algorithms (GAs) are search and optimization algorithms which are based on the fundamentals of natural evolution. In Fig. 1 we represent Evolutionary Algorithms in relation to other search techniques with special attention directed to Genetic Algorithms.

The basic principles of GAs were first presented by Holland [9]. Since that time GAs have been successfully applied to a wide range of problems including multimodal function optimization, machine learning, and the evolution of complex structures such as neural networks. An overview of GAs and their implementation in various fields is given by Goldberg [7] or Michalewicz [11].

When applying GAs to large and complex problems, one of the most frequent difficulties is premature convergence. Roughly speaking, premature convergence occurs when the population of a GA reaches such a suboptimal state that the

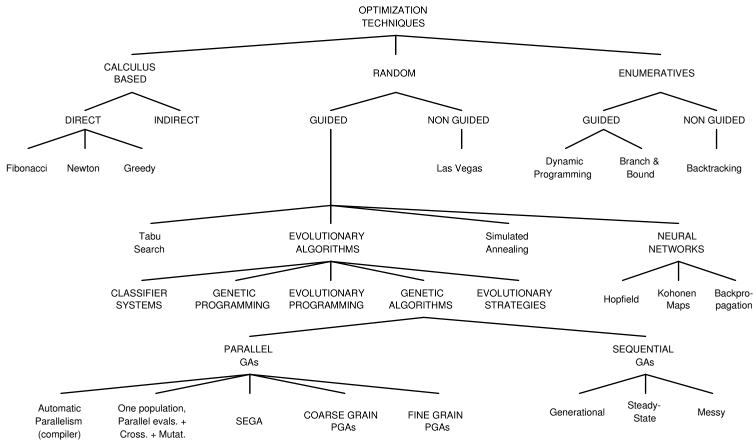

Fig. 1. Taxonomy of optimization techniques.

genetic operators (crossover, mutation) can no longer produce offspring that outperform their parents. Various methods have been proposed to retard the unwanted effects of premature convergence. Among others, these include modifications in the recombination procedure, in the selection procedure or in the fitness assignment (e.g. [15]). However, the effects of all these methods vary with different problems and their implementations.

A critical problem when studying premature convergence is the identification of its occurrence and its extent. In some contributions the difference between the average and maximum fitness is used as a value to measure premature convergence and the crossover and mutation probabilities are varied adaptively according to this measurement. Also the term population diversity has been used in many papers (e.g. [14]) to study premature convergence where the decrease of population diversity is considered as the primary reason for premature convergence. Therefore, a very homogeneous population, i.e. little population diversity, is the major reason for a Genetic Algorithm to prematurely converge. However, so far there exists little effort in performing a generic analysis of population diversity. But it is well known that the degree of population diversity is controllable by selective pressure.

In case of standard GAs selective pressure can only be influenced by the population size, by the choice of the selection mechanism and the operators as well as by the corresponding parameters. As these controls are quite complicated and also influence other characteristics of the GA we have modified the basic concept of GAs in a way that allows a simple and direct steering of selective pressure with only one additional parameter ([2], [4]). In a way quite similar to the (μ, λ) Evolution Strategy (ES) a virtual population of a size not smaller than the actual population size is introduced. Like in a standard GA the individuals of this virtual population are generated by selection, crossover and mutation. The actual new generation, i.e. the population that provides the heritable information for the remaining search process, is then built up with the best members of the virtual population. The greater the virtual population size is adjusted in comparison to the actual population size, the higher is the actual setting of selective pressure. In case of an equal sized virtual and actual population the advanced algorithm operates completely analogical with practically the same running time as the underlying GA. With this enhanced model it is already possible to achieve results clearly superior to the results of a comparable GA. Furthermore, this model allows a natural and intuitive formulation of further new biologically inspired parallel hybrid GA approaches ([1], [3]).

Even if these new GA variants are able to significantly outperform comparable GAs and similar heuristic optimization techniques in terms of global solution quality, a major drawback is the time-consuming job of parameter tuning. As GAs implement the idea of evolution, and as evolution itself must have evolved to reach its current state of sophistication, it is only natural to expect adaptation not only to be used for finding solutions to a problem, but also for tuning the algorithm to the particular problem. Therefore, this paper discusses new generic aspects of self-adaptation - especially for selective pressure steering.

In doing so we adopt the $\frac{1}{5}$ success rule postulated by Rechenberg for Evolution Strategies into our enhanced selection model for GAs. With this strategy it becomes possible to steer the evolutionary search process in such a way that sufficiently enough population diversity is maintained for the GA not to prematurely converge without the need to set up additional problem and implementation specific schedules for selective pressure steering as required in the implementations discussed in [2] and [4]. Furthermore, the algorithm is now able to automatically detect the phase when premature convergence effectively occurs which gives a reasonable termination criterion for our enhanced GA and, even more important, this self adaptive strategy for selective pressure steering can be used as a detector for an appropriate reunification date of subpopulations for the massively parallel SEGA-algorithm ([1], [3]). Thus, these newly introduced aspects of self adaptation make the algorithm more user-friendly and save a lot of testing work and parameter adjustment when applying the certain GA-derivatives to various kinds of problems. The experimental part of the paper compares the obtained results to the results of a corresponding standard GA.

2 The Self-Adaptive Selection Model

The basic idea of Evolutionary Algorithms is to merge the genetic information in a way that those building blocks will 'survive' during the evolutionary process which are essential for a global solution w.r.t. a given fitness function. In this context, the aim of selection is to choose those candidates for reproduction which are rather expected to contain the essential building-blocks. All popular selection mechanisms like roulette wheel, linear-ranking, or tournament selection fulfill this essential requirement, where the main difference of these strategies is given by their diverse selection pressure as described in [6]. However, it is a common property of all mentioned selection schemes that they consider only the fitness value of the parents which are chosen for reproduction. Therefore, parents with above-average fitness values are selected for crossover with higher probability - but the quality of the children who are generated from the selected parents is not taken into account. Especially in case of artificial evolution it happens quite frequently that essential building blocks of the genetic information in the parent generation get lost in the reproduction process and are therefore no more available for the ongoing evolutionary process.

Inspired by Rechenberg's $\frac{1}{5}$ success rule for Evolution Strategies (e.g. described in [13]), we have developed an advanced selection model for Genetic Algorithms that allows self-adaptive control of selection pressure in a quite intuitive way:

In a first (conventional) selection step, parents are chosen for reproduction by roulette-wheel, liner-rank or some kind of tournament-selection like in the case of a normal Evolutionary Algorithm. The difference to conventional selection is that the offspring generated from the selected parents do not automatically become members of the next generation. In our new model the quality of reproduction is measured in a second step of selection by comparing the fitness of the child with the fitness values of its parents. Only if the fitness value of the generated child is 'better' than the fitness of its parents, the child is accepted in the next generation, i.e. members of the mating pool for the further evolutionary process.

However, the meaning of 'better' has to be explained in more detail: is a child better than its parents, if it surpasses the fitness of the weaker, the better, or is it in fact some kind of mean value of both? For this problem we have decided to introduce a cooling strategy similar to Simulated Annealing. Following the basic principle of Simulated Annealing we claim that an offspring only has to surpass the fitness value of the worse parent in order to be considered as 'successful' at the beginning and while evolution proceeds the child has to be better than a fitness value continuously increasing between the fitness of the weaker and the better parent. Like in the case of Simulated Annealing, this strategy effects a broader search at the beginning whereas at the end of the search process this operator acts in a more and more directed way.

The number of offspring that have to be created in that way depends on a predefined ratio-parameter ($SuccRatio \in [0, 1]$) giving the quotient of next generation members that have to outperform their own(!!) parents. As long as this ratio is not fulfilled further children are produced. When the postulated ratio is reached, the rest of the next generation members are randomly chosen from the children that did not fulfill the fitness criterion. Within our new selection model we define selective pressure ($ActSelPress$) as the ratio of generated candidates to the population size. A default upper limit for selection pressure ($MaxSelPress$) gives a quite intuitive termination criterion: if it is no more possible to find a sufficient number of offspring that outperform their parents, premature convergence has occurred. Fig. 2 shows the operating sequence of the above described concepts.

By means of this novel selection strategy the appearance of clones is retarded and the bad properties of crossover operators are compensated - especially in

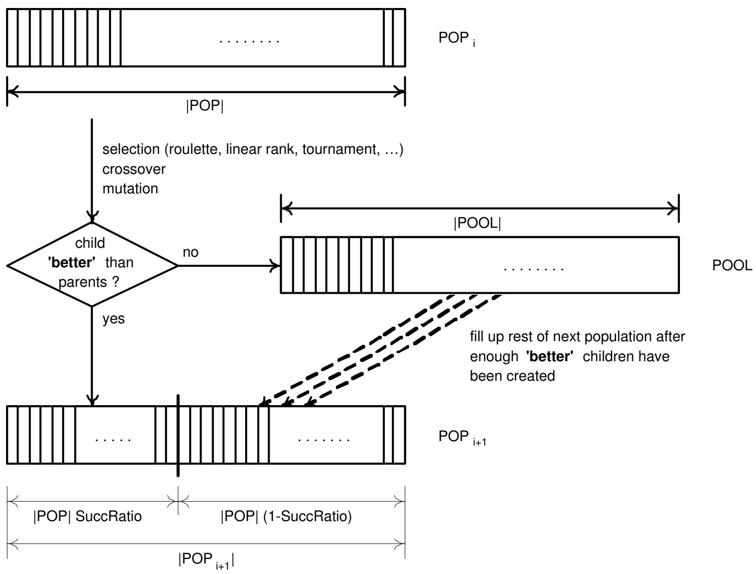

Fig. 2. Flowchart for embedding the new selection principle into a Genetic Algorithm.

case of rather critical crossover mechanisms as often applied in practical applications. As crossover results which do not mix their parents' genetic information advantageously are simply not considered, the ongoing evolutionary process is indeed proceeded with building blocks mainly derived from the parent generation. Thus, evolutionary search can be directed more efficiently without supporting the unwanted effects of premature convergence caused by too uniform solution candidates.

An additionally introduced concept suggests the appliance of multiple crossover operators simultaneously in order to roughly imitate the parallel evolution of a variety of species. This strategy seems very suitable for problems which consider more than one crossover operator - especially if the properties of the available operators may change as evolution proceeds. Furthermore, it is observable that the maintenance of population diversity is supported if more crossover operators are involved.

As an important property of the newly introduced methods it has to be pointed out that the corresponding GA is unrestrictedly included in this new variant of an Evolutionary Algorithm under special parameter settings. The experimental part analyzes the new algorithm for the Traveling Salesman Problem (TSP) which represents a very well documented instance of a multimodal combinatorial optimization problem. In contrast to all other evolutionary heuristics known to the authors that do not use any additional problem-specific information, we obtain solutions very close to the best-known solution for all considered benchmarks.

Algorithm 1 Standard Genetic Algorithm (SGA)

Initialize total number of iterations $nrOfIterations \in \mathbb{N}$
Initialize size of population $|POP|$
Produce an initial population POP_0 of size $|POP|$

for i = 1 to $nrOfIterations$ **do**
  Initialize next population POP_{i+1}

  while $|POP_{i+1}| \leq |POP|$ **do**
    Select two parents par_1 and par_2 from POP_i
    Generate a new child c from par_1 and par_2 by crossover
    Mutate c with a certain probability
    Insert c into POP_{i+1}
  end while
end for

Algorithm 2 Genetic Algorithm with New Selection

Initialize total number of iterations $nrOfIterations \in \mathbb{N}$
Initialize actual number of iterations $i = 0$
Initialize size of population $|POP|$
Initialize success ratio $SuccRatio \in [0, 1]$
Initialize maximum selection pressure $MaxSelPress \in [1, \infty[$
Initialize actual selection pressure $ActSelPress = 1$
Initialize comparison factor $CompFact = 0$
Produce an initial population POP_0 of size $|POP|$

while $(i < nrOfIterations) \wedge (ActSelPress < MaxSelPress)$ **do**
  Initialize next population POP_{i+1}
  Initialize pool for bad children $POOL$

  while $(|POP_{i+1}| < |POP| \cdot SuccRatio) \wedge (|POP_{i+1}| + |POOL| < |POP| \cdot MaxSelPress)$ **do**
    Generate a child from the members of POP_i due to their fitnesses by crossover and mutation

    Compare the fitness of the child c to the fitnesses of its parents par_1 and par_2 (w.l.o.g. assume that par_1 is fitter than par_2)
    if $f_c \leq f_{par_2} + |f_{par_1} - f_{par_2}| \cdot CompFact$ **then**
      Insert child into $POOL$
    else
      Insert child into POP_{+1}
    end if
  end while
  $ActSelPress = \frac{|POP_{i+1}|+|POOL|}{|POP|}$

  Fill up the rest of POP_{i+1} with members from $POOL$
  while $|POP_{i+1}| \leq |POP|$ **do**
    Insert a randomly chosen child from $POOL$ into POP_{i+1}
  end while
  Increase $CompFactor$ according to the used annealing strategy
  $i = i + 1$
end while

3 An Algorithmic Desription

Alg. 1 and Alg. 2 opposite the basic concept of a standard GA with the new GA that is equipped with the described self-adaptive selection mechanism in an algorithmic description. Even if the nomenclature 'Standard-GA' is not exactly standardized in GA literature we use this term for this version of a GA with which we qualitatively compare our new concepts.

4 Empirical Studies

Empirical studies with different problem classes and instances are widely considered as the most effective way to analyze the potential of heuristic optimization techniques like Evolutionary Algorithms. Even if a convergence proof similar to that of Simulated Annealing [8] may be possible, we are unfortunately confronted with the drawback that the number of states of the Markov Chain blows up from $|\mathcal{S}|$ to $|\mathcal{S}^{POP}|$, thus limiting the computational tractability to small problems and very small population sizes.

In our experiments, all computations are performed on a Pentium 4 PC with 1 GB of RAM. The programs are written in the programming language C#. For the tests we have selected the Traveling Salesman Problem (TSP) as a well documented instance of a typical multimodal combinatorial optimization problem. We have tested the new concepts on a selection of symmetric as well as asymmetric TSP benchmark problem instances taken from the TSPLIB [12] using updated results[1] for the best or at least best-known solutions. In all experiments, the results are represented as the relative difference to the best-known solution defined as $relativeDifference = (\frac{Result}{Optimal} - 1) \cdot 100\%$.

The first feature that is examined in the experimental part is the comparison of a GA with the new selection mechanism against the corresponding GA with conventional selection for a variety of crossover operators for the TSP [11], [10]. Tab. 2 shows these comparisons with roulette-wheel as the first selection step of our new model (respectively as the general selection mechanism in case of the standard-GA).

The evaluation of the results of Tab. 2 shows the remarkable effect that also crossover operators that are considered as rather unsuitable for the TSP [10] achieve quite good results in combination with the new selection model. The reason for this is given by the fact that in our selection-principle only children that have emerged from a good combination of their parents' attributes are considered for the further evolutionary process when the success ratio is set to a high value. In combination with a high upper value for the maximum selection pressure genetic search can therefore be guided advantageously also for poor crossover operators, as the larger amount of handicapped offspring is simply not considered for the further evolutionary process.

[1] Updates for the best-(known) solutions can be found for example on http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/

Table 1. Parameter set used in the model of Table 2.

generations	5000
population size	200
mutation rate	0.05
success ratio	0.6

Table 2. Comparison of SGA against the GA with new selection.

		SGA		GA with new selection		
Problem	Crosover	Best	Average	Best	Average	Change
eil76	OX	2.79	5.70	2.58	3.19	2.51
eil76	OBX	57.81	69.89	14.13	24.23	45.66
eil76	ERX	80.86	82.03	2.79	4.03	78.00
eil76	PMX	14.87	19.14	7.25	10.84	8.30
eil76	CX	22.86	37.05	8.74	11.09	25.96
eil76	MPX	119.70	131.23	2.60	3.66	127.57
ch130	OX	7.69	10.21	7.66	9.95	0.26
ch130	OBX	185.17	202.50	70.44	90.81	111.69
ch130	ERX	220.31	238.45	6.22	8.13	230.32
ch130	PMX	55.84	56.62	9.79	11.34	45.28
ch130	CX	122.49	154.19	12.29	14.43	139.76
ch130	MPX	223.72	240.59	3.27	4.89	235.70
kroA200	OX	24.67	27.75	8.97	19.69	8.06
kroA200	OBX	362.59	389.99	100.75	175.07	214.92
kroA200	ERX	481.1	490.99	35.60	51.69	439.30
kroA200	PMX	170.51	208.11	18.47	30.23	177.88
kroA200	CX	267.98	352.01	19.82	24.46	327.54
kroA200	MPX	365.24	392.73	9.71	11.92	380.81
ftv55	OX	22.95	28.13	19.53	22.55	5.58
ftv55	OBX	32.28	50.85	12.25	37.96	12.89
ftv55	ERX	89.55	93.45	13.74	17.58	75.87
ftv55	PMX	23.07	27.07	25.57	30.67	-3.60
ftv55	CX	61.69	65.36	64.27	67.42	-2.06
ftv55	MPX	112.75	124.56	0.00	0.89	123.67

For reasons of comparability of the results the parameters are set to the same values for both, the standard GA as well as for the GA with enhanced selection. The parameter settings for the test-runs shown in Tab. 2 for the standard GA as well as for the GA with the additional selection step are shown in Tab. 1.

5 Conclusion

The enhanced selection mechanism presented in this paper combines aspects of Evolution Strategies (selection pressure, success rule) and Simulated Annealing (growing selection pressure) with crossover and mutation of the general model of Genetic Algorithms. Therefore, established crossover and mutation operators for certain problems may be used analogously to the corresponding Genetic Algorithm. The investigations in this paper mainly focus on the avoidance of premature convergence and on the improvement of proven bad crossover operators.

Under special parameter settings the corresponding Genetic Algorithm is entirely included within the introduced concepts achieving an execution time only marginally worse than the execution time of the equivalent Genetic Algorithm. In other words, the introduced models can be interpreted as a generic extension of the GA-model. Therefore, an implementation of the new algorithm for a certain problem should be quite easy to do, presumed that the corresponding Genetic Algorithm (coding, operators) is known.

Especially in practical applications of Evolutionary Algorithms, where the capability of the introduced crossover operators is often not analyzed, the proposed selection should allow remarkable improvements of the solution quality. It is also believed by the authors that the use of the introduced selection should cause significant improvements when being applied to Genetic Programming applications because crossover mechanisms of Genetic Programming tend to produce offspring that do not combine the favorable properties of their parents. Furthermore, the presented selection operator should be very easily adoptable for a steady-state Genetic Algorithm and it would surely be an interesting research topic to analyze this interaction.

Based upon this newly postulated basic selection principle the mechanisms can also be combined with the already proposed Segregative Genetic Algorithm (SEGA) [3], an advanced Genetic Algorithm that introduces parallelism mainly to improve global solution quality. As a whole, a new generic evolutionary algorithm (SASEGASA) is introduced. A preliminary version of this algorithm is discussed in [5].

References

1. Affenzeller, M.: A New Approach to Evolutionary Computation: Segregative Genetic Algorithms (SEGA). Connectionist Models of Neurons, Learning Processes, and Artificial Intelligence, Lecture Notes of Computer Science 2084 (2001) 594–601

2. Affenzeller, M.: Transferring the Concept of Selective Pressure from Evolutionary Strategies to Genetic Algorithms. Proceedings of the 14th International Conference on Systems Science 2 (2001) 346–353
3. Affenzeller, M.: Segregative Genetic Algorithms (SEGA): A Hybrid Superstructure Upwards Compatible to Genetic Algorithms for Retarding Premature Convergence. International Journal of Computers, Systems and Signsls (IJCSS) Vol.2 No.1 (2001) 18–32
4. Affenzeller, M.: A Generic Evolutionary Computation Approach Based Upun Genetic Algorithms and Evolution Strategies.Journal of Systems Science Vol.28 No.4 (2002)
5. Affenzeller, M., Wagner, S.: SASEGASA: An Evolutionary Algorithm for Retarding Premature Convergence by Self-Adaptive Selection Pressure Steering. Accepted for IWANN 2003, Lecture Notes of Computer Science (2003)
6. Baeck, T.: Selective Pressure in Evolutionary Algorithms: A Characterization of Selection Mechanisms. Proceedings of the First IEEE Conference on Evolutionary Computation 1994: (1993) 57–62
7. Goldberg, D. E.: Genetic Alogorithms in Search, Optimization and Machine Learning. Addison Wesley Longman (1989)
8. Hajek, B.: Cooling Schedules for Optimal Annealing. Operations Research 13 (1988) 311–329
9. Holland, J. H.: Adaption in Natural and Artificial Systems. 1st MIT Press ed. (1992)
10. Larranaga, P., Kuijpers, C.M.H., Murga, R.H., Inza, I., Dizdarevic, S.: Genetic Algorithms for the Travelling Salesman Problem: A Review of Representations and Operators. Artificial Intelligence Review 13 (1999) 129–170
11. Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Programs. 3rd edn. Springer-Verlag, Berlin Heidelberg New York (1996)
12. Reinelt, G.: TSPLIB - A Traveling Salesman Problem Library. ORSA Journal on Computing 3 (1991) 376-384
13. Schoeneburg, E., Heinzmann, F., Feddersen, S.: Genetische Algorithmen und Evolutionsstrategien. Addison-Wesley (1994)
14. Smith, R.E. et al.: Population Diversity in an Immune System Model: Implications for Genetic Search. Foundations of Genetic Algorithms 2 (1993) 153–166
15. Srinivas M. et al.: Adaptive probabilities of crossover and mutation in genetic algorithms. IEEE Transactions on Systems, Man, and Cybernetics Vol.24 No.4 (1994) 656–667