SASEGASA: An Evolutionary Algorithm for Retarding Premature Convergence by Self-Adaptive Selection Pressure Steering

Michael Affenzeller, Stefan Wagner

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

Abstract. This paper presents a new generic Evolutionary Algorithm (EA) for retarding the unwanted effects of premature convergence. This is accomplished by a combination of interacting methods. To be intent on this a new selection scheme is introduced, which is designed to maintain the genetic diversity within the population by advantageous self-adaptive steering of selection pressure. Additionally this new selection model enables a quite intuitive condition to detect premature convergence. Based upon this newly postulated basic principle the new selection mechanism is combined with the already proposed Segregative Genetic Algorithm (SEGA) [3], an advanced Genetic Algorithm (GA) that introduces parallelism mainly to improve global solution quality. As a whole, a new generic evolutionary algorithm (SASEGASA) is introduced. The performance of the algorithm is evaluated on a set of characteristic benchmark problems. Computational results show that the new method is capable of producing highest quality solutions without any problem-specific additions.

1 Introduction

Evolutionary Algorithms (EAs) may be described as a class of bionic techniques that imitate the evolution of a species. The most important representatives of EAs are Genetic Algorithms (GAs) and Evolution Strategies (ES).

The fundamental principles of GAs were first presented by Holland [6]. 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 [5] or Michalewicz [9].

Evolution Strategies, the second major representative of EAs, were introduced by Rechenberg [10] and Schwefel [13]. Applied to problems of combinatorial optimization, ES tend to find local optima quite efficiently. Though, in the case of multimodal test functions, global optima can be detected by ES only if one of the starting values is located in the absorbing region of a global optimum.

The advantage of applying GAs to hard problems of combinatorial optimization lies in the ability to search the solution space in a broader way than heuristic methods based upon neighborhood search. Nevertheless, also GAs are frequently faced with a problem which, at least in its impact, is quite similar to the problem of stagnating in a local but not global optimum. This drawback, called premature convergence in the terminology of GAs, occurs when the population of a GA reaches such a suboptimal state that the genetic operators can no longer produce offspring that outperform their parents (e.g. [4]).

Inspired by Rechenberg's 1/5 success rule for Evolution Strategies, we have developed an advanced selection model for Genetic Algorithms that allows self-adaptive control of selection pressure in a quite intuitive way. Based upon this enhanced EA-model further generic extensions are being discussed.

The experimental part analyzes the new algorithms for the Traveling Salesman Problem (TSP) as 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 the best known solution for all considered benchmarks.

2 The Self-Adaptive Selection Model

As there exists no manageable model for a controllable handling of selection pressure within the theory of GAs[12], we have introduced some kind of intermediate step (a 'virtual population') into the selection process which provides a handling of selection pressure very similar to that of ES [2]. As we have exemplarily pointed out in contribution [2], the most common replacement mechanisms can easily be implemented in this intermediate selection step. Furthermore, this Evolution Strategy like variable selection pressure supported us to steer the degree of population diversity. However, within this model it is necessary to adjust a parameter for the actual selection pressure and in order to steer the search process advantageously a lot of parameter tuning is essential.

Motivated by those considerations we have set up an advanced selection model by introducing a new criterion abutted on Rechenberg's 1/5 success rule. The first selection step chooses the parents for crossover in the well-known way of Genetic Algorithms by roulette wheel, linear-rank, or some kind of tournament selection strategy. After having performed crossover with the selected parents we introduce a further selection mechanism that considers the success of the applied crossover in order to assure the proceeding of genetic search mainly with successful offspring in that way that the used crossover operator was able to create a child that surpasses its parents' fitness.

In doing so, a new parameter, called success ratio ($SuccRatio \in [0, 1]$), gives the quotient of the next population members that have to be generated by successful mating in relation to the total population size. Our adaptation of Rechenberg's success rule for GAs says that a child is successful if its fitness is better than the fitness of its parents, whereby 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 some kind of mean value of both? For this problem we have decided to introduce a Simulated Annealing (SA) like cooling strategy. Following the basic principle of SA we claim that a successful descendent has to surpass the fitness value of the worse parent at the beginning and while evolution proceeds the child has to be better than a fitness value continuously increasing in the range between the fitness of the weaker and the better parent. Like in the case of SA 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. Having filled up the claimed ratio ($SuccRatio$) of the next generation with successful individuals in the above meaning we simply fill up the rest of the next generation ($(1-SuccRatio)*|POP|$) with individuals arbitrarily chosen from the pool of individuals that were also created by crossover but did not reach the success critereon. The actual selection pressure $ActSelPress$ at the end of a single generation is defined by the quotient of individuals that had to be considered until the success ratio was reached and the number of individuals in the population $ActSelPress = \frac{|virtualPOP|+SuccRatio*|POP|}{POP}$. Fig. 1 shows the operating sequence of the above described concepts.

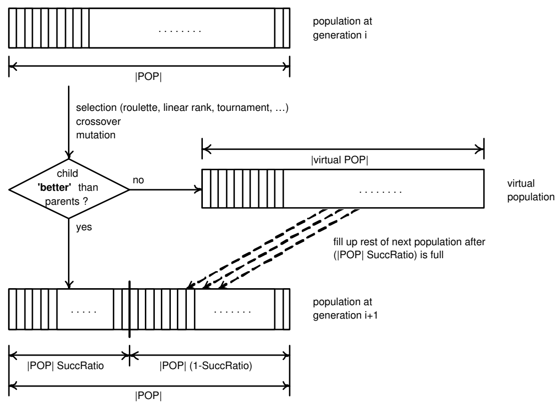

Fig. 1. Flowchart for embedding the new selection principle into a GA

With an upper limit of selection pressure ($MaxSelPress$), defining the maximum number of children considered for the next generation that may be produced in order to fulfill the success ratio, this new model also acts as a precise detector of premature convergence: If it is not possible anymore to find enough ($SuccRatio * |POP|$) children outperforming their own parents, even if ($MaxSelPress*|POP|$) candidates have been generated premature convergence has occurred.

As a basic principle of this selection model a higher success ratio causes higher selection pressure. Nevertheless, higher settings of success ratio and therefore of selection pressure do not essentially cause premature convergence. This is because per definition the new selection step (after crossover) does not accept clones that emanate from two identical parents. In traditional GAs such clones represent a major reason for premature convergence of the whole population around a suboptimal value, whereas the new selection step specifically counteracts against this phenomenon.

Experiments performed on the variable selection pressure model already indicate the supremacy of this approach. Also the corresponding canonical Genetic Algorithm is fully included in our new superstructure if the success ratio is simply set to 0. Furthermore, we are going to discuss new aspects and models built up upon the described self adaptive selection model In the following section.

3 Generic GA-Concepts Based Upon the Self-Adaptive Selection Model

When applying Genetic Algorithms to complex problems, one of the most frequent difficulties is premature convergence. Concisely speaking, premature convergence occurs when the population of a Genetic Algorithm reaches such a suboptimal state that the genetic operators can no longer produce offspring that outperform their parents (e.g. [4]).

A critical problem in studying premature convergence is the identification of its occurrence and the measure of its extent. Srinivas and Patnaik [15], for example, use the difference between the average and maximum fitness as a standard to measure premature convergence and adaptively vary the crossover and mutation probabilities according to this measurement. On the other hand, as in the present paper, the term 'population diversity' has been used in many papers to study premature convergence (e.g. [14]) where the decrease of population diversity is considered as the primary reason for premature convergence.

The following generic extensions, that are built up upon the self-adaptive variable selection pressure model, aim to retard premature convergence in a general way.

3.1 SASEGASA: The Core Algorithm

In principle, the new SASEGASA (Self Adaptive SEgregative Genetic Algorithm with Simulated Annealing aspects) introduces two enhancements to the basic concept of Genetic Algorithms. Firstly, we bring in a variable selection pressure, as described in section 2, in order to control the diversity of the evolving population. The second concept introduces a separation of the population to increase the broadness of the search process and joins the subpopulation after their evolution in order to end up with a population including all genetic information sufficient for locating a global optimum.

The aim of dividing the whole population into a certain number of subpopulations (segregation) that grow together in case of stagnating fitness within those subpopulations (reunification) is to combat premature convergence which is the source of GA-difficulties. This segregation and reunification approach is an efficient method to overcome premature convergence [1] called the SEGA algorithm (SEgregative GA).

The principle idea of SEGA is to divide the whole population into a certain number of subpopulations at the beginning of the evolutionary process. These subpopulations evolve independently from each other until the fitness increase stagnates in all subpopulations because of too similar individuals within the subpopulations, i.e. local premature convergence. Then a reunification from n to $(n-1)$ subpopulations is performed by joining an appropriate number of adjacent subpopulation members.

Metaphorically speaking this means, that the a certain number of villages (subpopulations) at the beginning of the evolutionary process are slowly growing together to bigger cities, ending up with one big town containing the whole population at the end of evolution. By this approach of width-search essential building blocks can evolve independently in different regions of the search space at the beginning and during the evolutionary process. In the case of a standard GA those building blocks are likely to disappear early and, therefore, their genetic information can not be provided at a later phase of evolution, when the search for a global optimum is of paramount importance.

Within the classical SEGA algorithm there is no criterion to detect premature convergence and there is also no self-adaptive selection pressure steering mechanism. Even if the results of SEGA are quite good with regard to global convergence it requires an experienced user to adjust the selection pressure steering parameters and as there is no criterion to detect premature convergence the dates of reunification have to be implemented statically.

Equipped with our new self adaptive selection technique we have both: A self-adaptive selection pressure (depending on the given success ratio) as well as an automated detection of local premature convergence, if the current selection pressure becomes higher then the given maximal selection pressure parameter ($MaxSelPress$). Therefore, a date of reunification has to be set, if local premature convergence has occurred within all subpopulations in order to increase genetic diversity again.

Again, like in the context of the new selection model which is included in SASEGASA as well, it should be pointed out that a corresponding Genetic Algorithm is unrestrictedly included in SASEGASA, when the number of subpopulations (villages) is set to 1 and the success ratio is set to 0 at the beginning of the evolutionary process. Moreover, the introduced techniques also do not use any problem specific information.

4 Experimental Results

Empirical studies with different problem classes and instances are the most effective way to analyze the potential of heuristic optimization searches like Evolutionary Algorithms.

In our experiments, all computations are performed on a Pentium 4 PC with 1 GB RAM. The programs are written in the C# programming language. 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 [11] using updated results for the best or at least the best known solutions. In all experiments, the results are represented as the relative difference to the best known solution defined as relativeDifference $= (\frac{Result}{Optimal} - 1) * 100\%$.

Especially we aim to point out the main effect of the present contribution - namely that an increasing number of subpopulations at the beginning of the evolutionary process allows to achieve scalable improvements in terms of global convergence. As Tab. 1 shows, the global solution can be scaled up to highest quality by just increasing the number of evolving subpopulations. This definitely represents a new achievement in the area of Evolutionary Algorithms with distributed subpopulations. For all the experiments in Tab. 1 the starting size of one subpopulation is fixed at a value of 100, mutation rate is 5%, and the upper limit of selection pressure is set to a value of 10 for the TSPs respectively 15 for the ATSPs.

The results in Tab. 1 present the best solution quality of five runs of each test-instance as well as the average best result-value of the five runs in terms of average difference to the best known solution.

Indeed, as Tab. 1 shows, the optimal solution could be found for all benchmark test cases if the initial number of subpopulations is set high enough.

Even if the achieved results are clearly superior to most of the results reported for applications of Evolutionary Algorithms to the TSP [8], it has to be pointed out again, that all introduced and applied additions to a standard evolutionary algorithm are generic and absolutely no problem specific local pre- or post-optimization techniques have been applied in our experiments. Additional experiments performed on non-standardized scheduling problems (job-shop and multiprocessor) show comparable potential and underscore the generic potential of the new techniques in various fields of applications.

5 Conclusion

In this paper an enhanced Genetic Algorithm and two upgrades have been presented and exemplarily tested on some TSP benchmarks. The proposed EA-based techniques couple aspects from Evolution Strategies (selection pressure and success rule in our new selection procedure), Simulated Annealing (growing selective pressure) as well as a special segregation and reunification strategy with crossover and mutation in the general model of a Genetic Algorithm. Therefore, established crossover and mutation operators for certain problems may be used analogously to the corresponding Genetic Algorithm. The investigations in this paper have mainly focused on the avoidance of premature convergence and on

Table 1. Experimental results of the SASEGASA-algorithm for TSPLIB benchmark problems with variable number of subpopulations for tuning global solution quality.

Problem	Parameters and Operators				Results (in %)	
	$noOfSubPopulations$	SuccessRatio	Crossover	Iterations	Best	Average
berlin52	1	0,8	OX,ERX,COSA	139	6.7	8.6
berlin52	5	0,8	OX,ERX,COSA	218	0.0	0.0
berlin52	10	0,8	OX,ERX,COSA	301	0.0	0.0
ch130	1	0,8	OX,ERX,COSA	295	52.9	59.4
ch130	5	0,8	OX,ERX,COSA	584	12.2	13.24
ch130	10	0,8	OX,ERX,COSA	783	4.8	6.11
ch130	20	0,8	OX,ERX,COSA	1024	1.6	2.5
ch130	40	0,8	OX,ERX,COSA	1426	0.63	0.89
ch130	80	0,8	OX,ERX,COSA	2067	0.45	0.74
ch130	160	0,8	OX,ERX,COSA	3518	0.0	0.15
kroA200	1	0,8	OX,ERX,COSA	584	77.6	87.3
kroA200	5	0,8	OX,ERX,COSA	1035	22.5	24.9
kroA200	10	0,8	OX,ERX,COSA	1310	12.4	13.1
kroA200	20	0,8	OX,ERX,COSA	1604	4.7	7.4
kroA200	40	0,8	OX,ERX,COSA	2243	0.9	2.6
kroA200	80	0,8	OX,ERX,COSA	2842	0.6	1.3
kroA200	160	0,8	OX,ERX,COSA	4736	0.0	0.3
rbg323	1	0,8	OX,ERX,COSA	1463	20.59	26.24
rbg323	5	0,8	OX,ERX,COSA	2690	4.37	8.02
rbg323	10	0,8	OX,ERX,COSA	5991	2.11	2.84
rbg323	20	0,8	OX,ERX,COSA	13456	0.30	0.53
rbg323	40	0,8	OX,ERX,COSA	40762	0.15	0.20
rbg323	80	0,8	OX,ERX,COSA	100477	0.00	0.06
rbg323	160	0,8	OX,ERX,COSA	212418	0.00	0.00

the introduction of methods which make the algorithm more open for scalability in terms of solution quality versus computation time.

Concerning the speed of SASEGASA, it has to be pointed out that the superior performance concerning global convergence requires a higher computation time, mainly because of the greater total population size $|POP|$ and because of the increase of generated individuals in our new self adaptive selection mechanism under higher selection pressure. Nevertheless, in contrast to other implementations in the field of evolutionary computation, it is absolutely remarkable, that it has become possible to almost linearly improve the global solution quality by introducing greater population sizes and an accordingly greater number of subpopulations, whereas the computational costs are 'only' increasing linearly due to the greater number of individuals having to be taken into account. This allows to transfer already developed GA-concepts to increasingly powerful computer systems in order to achieve better results. Using parallel computer architectures seems especially suited to increase the performance of SASEGASA.

Anyway, with special parameter settings the corresponding Genetic Algorithm is fully included within the introduced concepts achieving a performance only marginally worse than the performance of the equivalent Genetic Algorithm. In other words, the introduced models can be interpreted as a superstructure of the GA model or as a technique upwards compatible to Genetic Algorithms. Therefore, an implementation of the new algorithm(s) for a certain problem should be quite easy to do, presumed that the corresponding Genetic Algorithm (coding, operators) is known.

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. Internatinal Journal of Computers, Systems and Signals (IJCSS), Vol. 2, Nr. 1 (2001) 18–32
4. Fogel, D.B.: An Introduction to Simulated Evolutionary Optimization. IEEE Trans. on Neural Networks 5(1) (1994) 3–14
5. Goldberg, D. E.: Genetic Alogorithms in Search, Optimization and Machine Learning. Addison Wesley Longman (1989)
6. Holland, J. H.: Adaption in Natural and Artificial Systems. 1st MIT Press ed. (1992)
7. Kirkpatrick, S., Gelatt Jr., C.D., Vecchi, M.P.: Optimization by Simulated Annealing. Science 220 (1983) 671–680
8. 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
9. Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Programs. 3rd edn. Springer-Verlag, Berlin Heidelberg New York (1996)
10. Rechenberg, I.: Evolutionsstrategie. Friedrich Frommann Verlag (1973)
11. Reinelt, G.: TSPLIB - A Traveling Salesman Problem Library. ORSA Journal on Computing 3 (1991) 376-384
12. Schneburg, E., Heinzmann, F., Feddersen, S.: Genetische Algorithmen und Evolutionsstrategien. Addison-Wesley (1994)
13. Schwefel, H.-P.: Numerische Optimierung von Computer-Modellen mittels Evolutionsstrategie. Birkhäuser Verlag. Basel (1994)
14. Smith, R.E., Forrest, S., Perelson, A.S.: Population Diversity in an Immune System Model: Implications for Genetic Search. Foundations of Genetic Algorithms 2 (1993) 153–166
15. Srinivas, M., Patnaik, L.: Adaptive Probabilities of Crossover and Mutation in Genetic Algorithms . IEEE Transactions on Systems, Man, and Cybernetics 24(4) (1994) 656–667