genetic algorithms, evolutionary strategies, selective pressure

Michael Affenzeller*

TRANSFERRING THE CONCEPT OF SELECTIVE PRESSURE FROM EVOLUTIONARY STRATEGIES TO GENETIC ALGORITHMS

Most problems of combinatorial optimization like routing, task allocation, or scheduling belong to the class of NP-complete problems and can be solved efficiently only by heuristics. Both, genetic algorithms and evolutionary strategies have a number of drawbacks that reduce their applicability to that kind of problems. In order to overcome some of these problems, this paper looks upon the standard genetic algorithm as an artificial self organizing process. With the purpose of providing concepts that make the algorithm more open for scalability on the one hand, and that fight premature convergence on the other hand, this paper presents an extension of the standard genetic algorithm that doesn't introduce any problem specific knowledge. On the basis of an evolutionary strategy like selective pressure some further improvements like the introduction of a concept to handle multiple crossover operators in parallel or the introduction of a concept of segregation and reunification of smaller subpopulations during the evolutionary process are considered. The additional aspects introduced within the scope of that variants of genetic algorithms are inspired from optimization as well as from the views of bionics. In contrast to contributions in the field of genetic algorithms that introduce new coding standards and operators for certain problems, the introduced approach should be considered as a heuristic appliable to multiple problems of combinatorial optimization using exactly the same coding standards and operators for crossover and mutation as done when treating a certain problem with a standard genetic algorithm. In the present paper the new algorithm and some of its variants are discussed for the travelling salesman problem (TSP) as a well documented instance of a multimodal combinatorial optimization problem.

1. INTRODUCTION

A genetic algorithm (GA) may be described as a mechanism that imitates the genetic evolution of a species. The underlying principles of GAs were first presented by Holland in [5]. An overview about GAs and their implementation in various fields have been given by Goldberg [4], Michalewicz [7] respectively in [2], and [3].

Evolutionary Strategies, the second major representative of evolutionary computation, were introduced by Rechenberg in [8].

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

Applied to problems of combinatorial optimization evolutionary strategies tend to find local optima quite efficiently. But in the case of multimodal test functions global optima can only be detected by evolutionary strategies if one of the start values is located in the narrower range of a global optimum. GAs on the other hand are mainly controlled by the crossover operator and therefore, a significant greater part of the search space is taken into account. That's why GAs are usually superior to evolutionary strategies in finding global optima of multimodal test functions. Nevertheless, the concept how evolutionary strategies handle the selective pressure has turned out to be very useful for the new GA and its variants as presented in this paper.

Motivated by that considerations we have developed an extended approach to GAs with the objective target to include some aspects of selection as used in the context of evolutionary strategies into the concept of GAs. Furthermore, we have borrowed the cooling mechanism from simulated annealing (SA), introduced by Kirkpatrick [6] in order to obtain a variable selective pressure. A further purpose in that stage of development of this new system was to keep it as problem independently and open to further adoptions as possible without loosing the possibility to use exactly the same operators for crossover and mutation as used when considering a certain problem with an ordinary GA. This strategy has the major advantage that well tried crossover and mutation operators for certain problems can further on be used within the scope of this new GA approach.

As a mutual basis a virtual population of adjustable size is introduced. The members of this virtual population are usually generated from the last population using the same crossover operator as used when treating the same problem with a usual GA, i.e. choosing two parents due to their fitness and creating an offspring. But moreover, this concept also allows a mixture of different crossover methods with dropping out descendants with lower fitness depending on how much the virtual population size exceeds the size of the population, i.e. if the size of the population is n and the size of the virtual population is 1.5*n, the best n candidates of the virtual generation – after mutation as a low probability event - are chosen as members of the new generation. Spoken in terms of evolution this means that a certain percentage of a population is not allowed to transfer its hereditary material into the next generation which represents a direct analogy to the most common variants of evolutionary strategies and allows to control the selective pressure of GAs in a very similar way as done in the context of evolutionary strategies.

In addition, this enhanced GA model allows further extensions that rely on a variable selective pressure as the general conditions may change during the evolutionary process. Two such cogitable extensions will be described in section 3.

Experimental results on some symmetric and asymmetric benchmark problems of the TSP indicate the supremacy of the introduced concepts for locating global minima compared to a standard GA. Furthermore, the evaluation shows, that the results of the segregative genetic algorithm (as introduced in subsection 3.1) are comparable for symmetric benchmark problems and even superior for asymmetric benchmark problems when being compared to the results of the cooperative simulated annealing technique (COSA) [12] which has to be considered as a problem specific heuristic for routing problems. This represents a major difference to the present approach that uses exactly the same operators as a corresponding GA and can, therefore, be applied to a huge number of problems - namely all problems GAs can be applied to.

Moreover it should be pointed out that the corresponding GA is unrestrictedly included in the presented variants of a genetic algorithm.

2. INTRODUCING A VARIABLE SELECTIVE PRESSURE INTO THE CONCEPT OF GENETIC ALGORITHMS

The handling of selective pressure in our context is mainly motivated by evolutionary strategies where μ parents produce λ descendants from which the best μ survive. Within the framework of evolutionary strategies, the selective pressure is defined as $s = \frac{\mu}{\lambda}$, where a small value of s indicates a high selective pressure and vice versa (for details see for instance [11]). Applied to the new genetic algorithm this means that from |POP| (population size) number of parents |POP|*T ((size of virtual population) >|POP|, i.e. T>1) descendants are generated by crossover and mutation from which the best |POP| survive as illustrated in Figure 1.

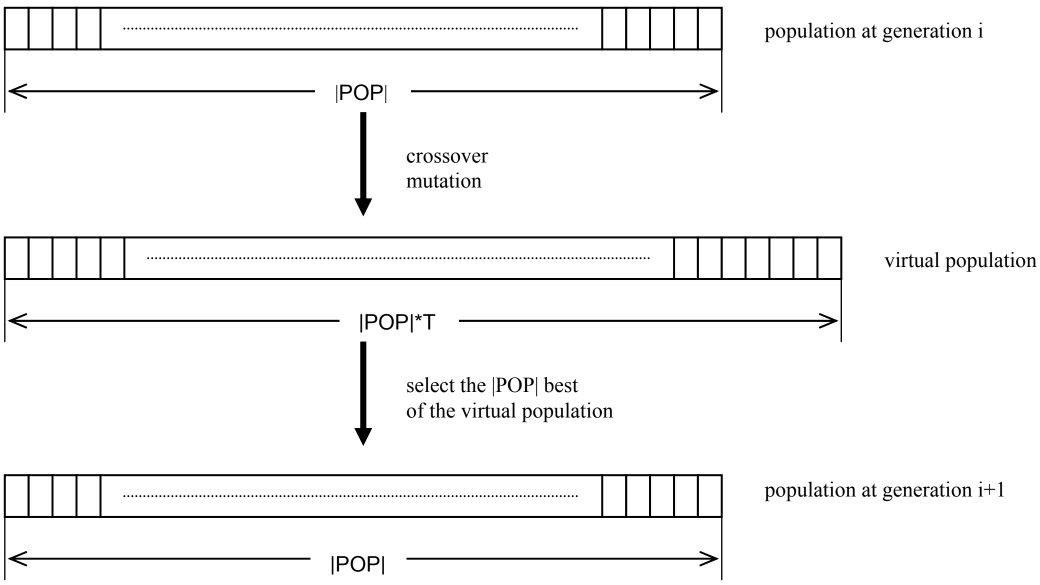

Figure 1. Evolution of a new population with selective pressure $s=\frac{1}{T}$.

Obviously we define the selective pressure as $s = \frac{|POP|}{|POP|*T} = \frac{1}{T}$ where a small value of s, i.e. a great value of T stands for a high selective pressure and vice versa. Equipped with this enhanced GA model it is quite easy to adopt further extensions based upon a controllable selective pressure, i.e. it becomes possible either to reset the temperature up/down to a certain level or simply to cool down the temperature in the sense of simulated annealing during the evolutionary process in order to steer the convergence of the algorithm.

3. NEW CONCEPTS BASED UPON THE VARIABLE SELECTIVE PRESSURE MODEL

In the following we will discuss two variants of a GA that use the variable selective pressure model as basis for further considerations:

3.1. SEGREGATIVE GENETIC ALGORITHMS

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 is to combat premature convergence which has to be considered as the source of GA-difficulties.

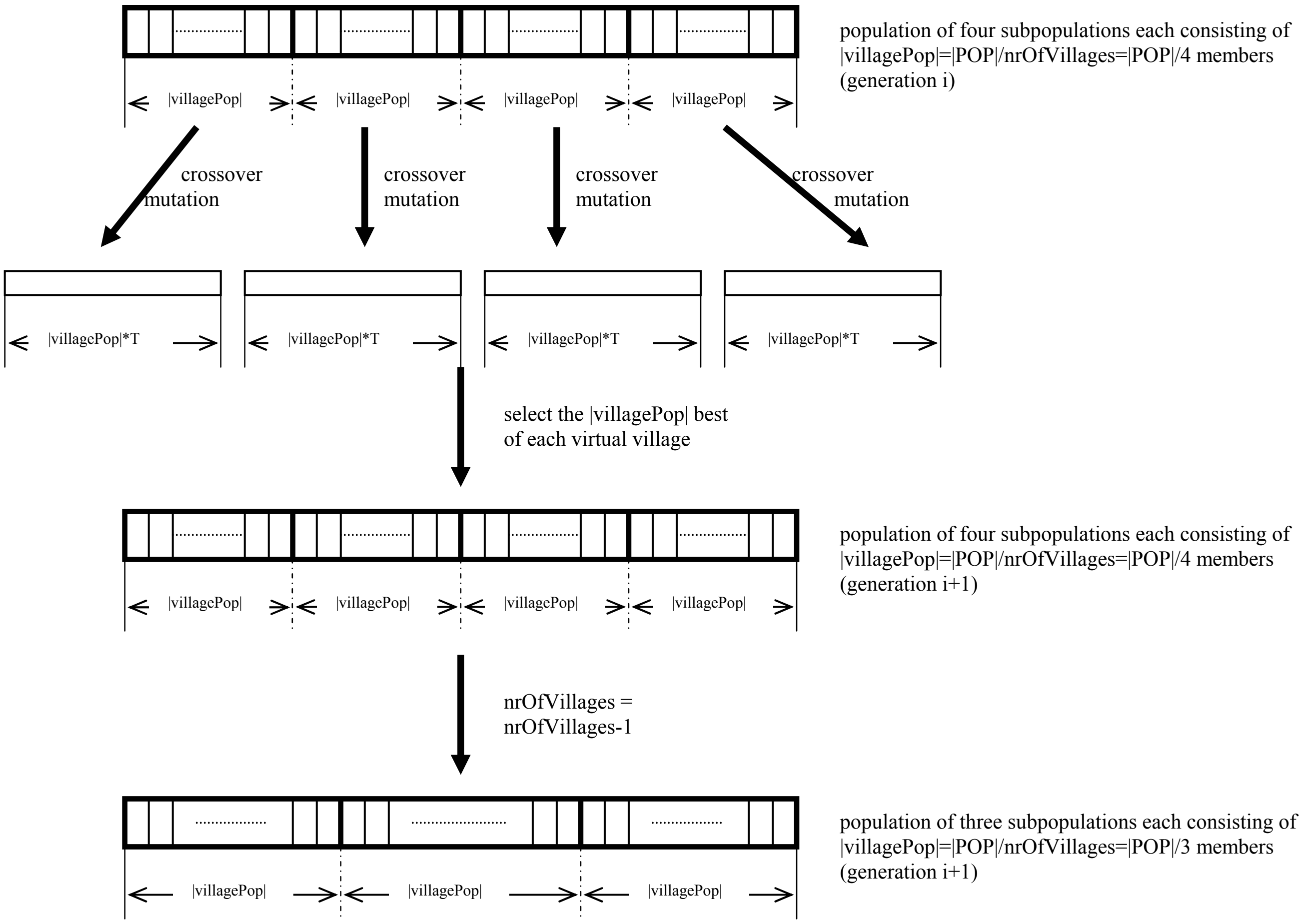

Figure 2. Evolution of a new population for the instance that four subpopulations are merged to three.

The principal idea 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 because of too similar individuals within the subpopulations. Then a reunification from n to (n-1) subpopulations is done. Figure 2 shows a schematic diagram of the described process. This process is repeated until all villages are growing together ending up in one town (reunification). By this approach of width-search, building blocks in different regions of the search space are evolved at the beginning and during the evolutionary process which would disappear early in case of standard GAs and whose genetic information could not be provided at a later date of evolution when the search for global optima is of paramount importance. The aim is that the best building-blocks survive during the recombination phase, yielding in a final population (if the number of villages is 1) containing all essential building-blocks for the detection of a global optimum. In case of ordinary GAs, building blocks that disappear early and which may be important at a later stage of the evolutionary process, when the search for global optima is of paramount importance, can hardly ever be reproduced (premature convergence).

In this context the above mentioned variable selective pressure is especially important when some residents of another village are joined to a certain village in order to steer the genetic diversity. For details on segregative genetic algorithms the reader is referred to [1].

3.2. THE USAGE OF MULTIPLE CROSSOVER OPERATORS IN PARALLEL

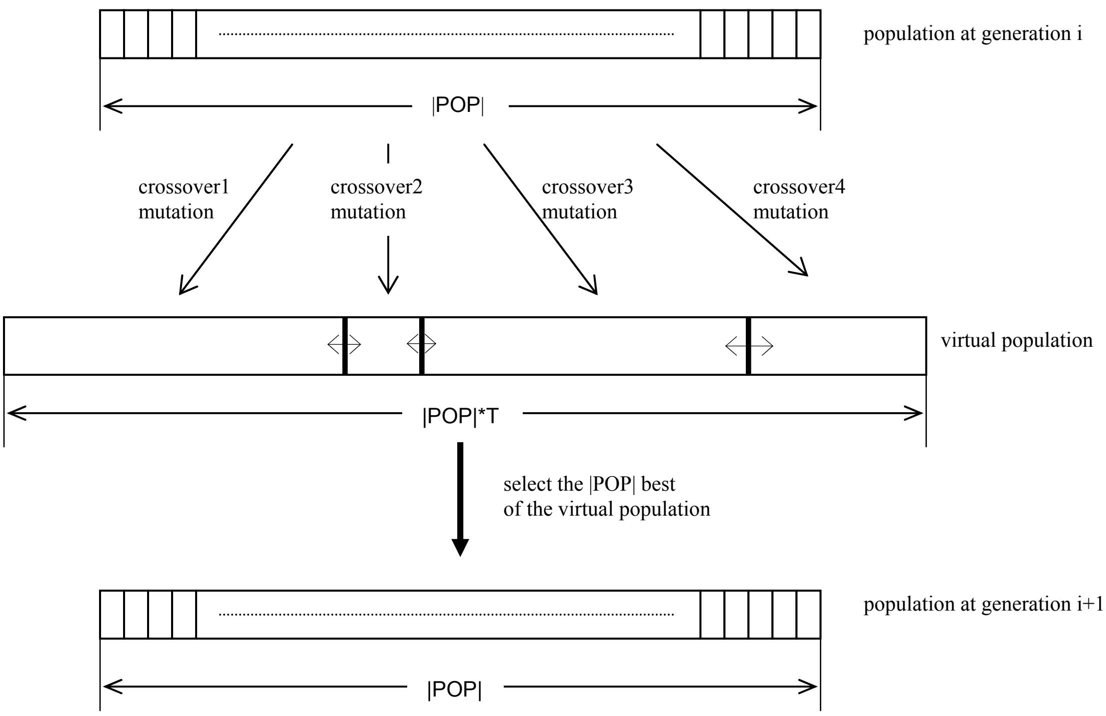

Figure 3. Evolution of a new population for the instance that four crossover operators are used where.

GAs as well as its most common variants consider the evolution of a single species, i.e. crossover can be done between all members of the population. This supports the aspect of depth-search but not the aspect of width-search. Considering the natural evolution, where a multiplicity of species evolve in parallel, as a role model we could introduce a number of crossover operators and apply each one to a certain subpopulation. In order to keep that model realistically it would be necessary to choose the size of those subpopulations dynamic, i.e. depending on the actual success of a certain species its living space is expanded or restricted. Speaking in the words of genetic algorithms, this means that the size of subpopulations (defined by the used crossover operator) with lower success in the sense of the quality function is restricted in support of those subpopulations that push the process of evolution.

But as no GA is able to model jumps in the evolutionary process and no exchange of information between the subpopulations takes place, the proposed strategy would fail in generating results superior to the results obtained when running the GAs with the certain operators one after another.

Therefore, it seems reasonable to allow also recombination of individuals that have emerged from different crossover operators, i.e. the total population is taken into account for each crossover operator and the living space of each virtual subpopulation is defined by its success during the last iterations as illustrated in Figure 3.

4. EXPERIMENTAL RESULTS

In our experiment, all computations are performed on a Pentium III PC with 256 megabytes of main memory. The programs are written in the Java programming language. We have tested two of the described new variants of a GA, namely a GA with a variable selective pressure equipped with an annealing like cooling strategy and SEGA as described is subsection 3.1. In doing so, we have used a selection of symmetric as well as asymmetric benchmark problem instances taken from the TSPLIB [9] using updated results for the best or at least the best known solutions taken from [10]. Furthermore, we have performed a comparison of our new results with a GA using exactly the same operators for crossover and mutation and the same parameter settings and with the COSA-algorithm as an established and successful ambassador of a heuristic especially developed for routing problems.

For the tests the parameters of COSA are set as suggested by the author in [12]. The GA and its scions use a mutation probability of 0.05 and a combination of OX-crossover [7] and ERX-crossover [12] combined with the golden-cage population model (e.g. [12]), i.e. the entire population is replaced with the exception that the best member of the old population survives until the new population generates a better one (wild-card strategy). The specific parameter settings of the GA with a variable selective pressure (temperature, α) and the specific parameter settings of SEGA (temperature, α, number of villages, dates of reunification) have been done by means of testing, nevertheless further parameter tuning should be possible.

Figure 4 shows the experimental results for the problem kro124p (100 city problem) as an example of an asymmetric TSP benchmark. This example demonstrates the predominance of the new SEGA compared to the standard-GA. Moreover it even shows the competitiveness of SEGA when compared to the problem specific COSA heuristic.

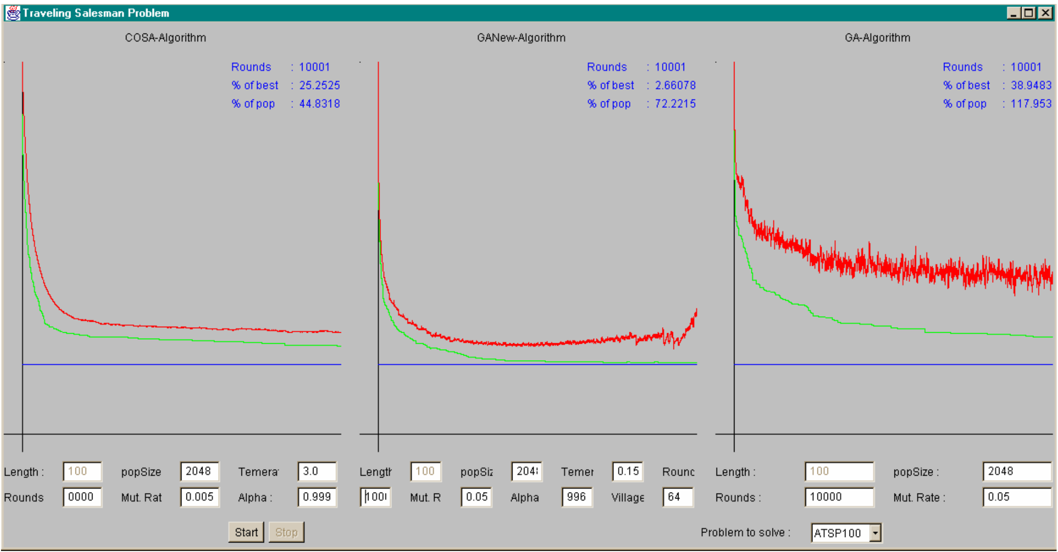

Figure 4. Comparison of COSA, SEGA, and GA on the basis of the kro124p benchmark problem: For each algorithm, the average fitness and the fitness of the best member of the population is diagrammed relatively to the best known solution represented by the horizontal line.

Table 1 shows the experimental results of COSA, GA, SEGA, and a GA with variable selective pressure concerning various types of problems in the TSPLIB. For each problem, the algorithms were run ten times. The efficiency for each algorithm is quantified in terms of the relative difference of the fitness of the best individual after a given number of iterations to the best or best known solution. In this experiment, the relative difference is defined as $relativeDifference = (\frac{Fitness}{Optimal} - 1) * 100\%$.

Problem	Nr.of iterations	Average Difference			
		COSA	GA	SEGA	GA+var.Sel.Pr.
eil76 (symmetric)	2000	3.22	11.21	1.55	4.21
ch130 (symmetric)	8000	4.76	35.44	1.84	8.91
kroA150 (symmetric)	8000	7.90	40.97	2.21	10.36
kroA200 (symmetric)	10000	8.54	45.11	5.21	14.77
br17 (asymmetric)	100	0.00	0.00	0.00	0.00
ftv55 (asymmetric)	2000	44.22	33.92	0.76	4.02
kro124p (asymmetric)	10000	26.78	37.49	2.61	20.06
ftv170 (asymmetric)	15000	202.33	131.61	4.13	73.55

Table 1. Experimental results of COSA, GA, SEGA and GA+variable selective pressure.

CONCLUSION

In this paper an enhanced genetic algorithm and two upgrades have been presented and exemplarily tested on some TSP benchmarks. The proposed GA-based techniques couple aspects from evolutionary strategies (selective pressure), simulated annealing (temperature, cooling) as well as a special segregation and reunification strategy with crossover, mutation and selection in a general way, so that established crossover and mutation operators for certain problems may be used analogous to the corresponding GA. The investigations in this paper have mainly focused on the avoidance of premature convergence and on the introduction of methods that make the algorithm more open for scalability in the sense of convergence versus running time.

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

However, the efficiency of a variable selective pressure certainly depends on the genetic diversity of the entire population and ongoing research indicates that it could be a very fruitful approach to define the actual selective pressure depending on the actual genetic diversity of the population.

REFERENCES

[1] AFFENZELLER M., *A New Approach to Evolutionary Computation: Segregative Genetic Algorithms (SEGA)*, to be published in IWANN'2001: Lecture Notes in Computer Science, Springer Verlag, Berlin Heidelberg New York 2001.

[2] CHAMBERS L. (ed.), *Practical Handbook of Genetic Algorithms Vol.I*, CRC Press 1995.

[3] CHAMBERS L. (ed.), *Practical Handbook of Genetic Algorithms Vol.II*, CRC Press 1995.

[4] GOLDBERG D.E., *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison Wesley Longman 1989.

[5] HOLLAND J.H., *Adaption in Natural and Artificial Systems*, 1st MIT Press ed. 1992.

[6] KIRKPATRICK S. et al., *Optimization by Simulated Annealing*, Science 220 1983, pp. 671-680.

[7] MICHALEWICZ Z., *Genetic Algorithms + Data Structures = Evolution Programs 3rd edn.*, Springer Verlag, Berlin Heidelberg New York 1996.

[8] RECHENBERG I., *Evolutionsstrategie*, Friedrich Frommann Verlag 1973.

[9] REINELT G., *TSPLIB – A Travelling Salesman Problem Library*, ORSA Journal on Computing 3 1991, pp. 376-384.

[10] REINELT G., *TSPLIB – A Travelling Salesman Problem Library*, ftp://ftp.zib.de/pub/Packages/mp-testdata/tsp/tsplib/index.html (1997).

[11] SCHÖNEBURG E. et al., *Genetische Algorithmen und Evolutionsstrategien.*, Addison Wesley 1994.

[12] WENDT O., *Tourenplanung durch Einsatz naturanaloger Verfahren*, Deutscher Universitätsverlag 1995.