genetic algorithms, evolution strategies, selective pressure

Michael Affenzeller*

A GENERIC EVOLUTIONARY COMPUTATION APPROACH BASED UPON GENETIC ALGORITHMS AND EVOLUTION STRATEGIES

Many problems that are treated by genetic algorithms belong to the class of NP-complete problems. The vantage of genetic algorithms when being applied to such kind of problems lies in the ability to search through the solution space in a broader sense than other heuristic methods that are based upon neighborhood search methods. Nevertheless, also genetic algorithms 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 solution what typically occurs when applying neighborhood based searches to hard problems with multimodal solution spaces. This drawback, called premature convergence in the terminology of genetic algorithms, 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. During the last decades plenty of work has been investigated to introduce new coding standards and operators in order to overcome this essential handicap of genetic algorithms. As these coding standards and the belonging operators are rather problem specific in general we try to take a different approach and look upon the concepts of genetic algorithms as an artificial self organizing process in a bionically inspired generic way in order to improve the global convergence behaviour of genetic algorithms independently of the actually employed implementation.

In doing so we have introduced an advanced selection model for genetic algorithms that allows adaptive selective pressure handling in a way that is quite similar to evolution strategies. This enhanced genetic algorithm-model allows further extensions 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. Both extensions rely on a variable selective pressure because the general conditions may change during the evolutionary process.

The experimental part of the paper discusses the new algorithms for the traveling salesman problem (TSP) as a well documented instance of a multimodal combinatorial optimization problem achieving results which significantly outperform the results obtained with a contrastable genetic algorithm.

1. INTRODUCTION

Work on what nowadays is called evolutionary computation started in the sixties in the United States and Germany. Substantially there are two basic approaches in computer science that copy evolutionary mechanisms: evolution strategies (ES) and genetic

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

algorithms (GA). Genetic algorithms go back to Holland [3], an American computer scientist and psychologist. Holland developed his theory not only under the aspect of solving optimization problems but also to observe biological processes. Last but not least that is the reason that genetic algorithms are much closer to the biological model than evolution strategies. The theoretical foundations of evolution strategies (ES) have been made by Rechenberg and Schwefel [7]. Their primary goal was optimization. Despite many things in common the two attempts developed almost independently from each other in the USA (GA) and Germany (ES).

Applied to problems of combinatorial optimization evolution strategies tend to find local optima quite efficiently. But in the case of multimodal test functions global optima can only be detected by evolution 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 evolution strategies in finding global optima of multimodal test functions. Nevertheless, the concept how evolution 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 evolutionary computation with the objective target to include some aspects of selection as used in the context of evolution strategies into the concept of GAs. Furthermore, we have borrowed the cooling mechanism from simulated annealing (SA), introduced by Kirkpatrick [4] 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. Depending on how much the virtual population size exceeds the size of the population this concept allows dynamic steering of selective pressure; 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 evolution strategies and allows to control the selective pressure of GAs in a very similar way as done in the context of evolution 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 4.

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 4.1) 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) [13] which has to be considered as a very efficient problem specific heuristic for routing problems.

2. GENERAL MOTIVATION

The considerations evolved in the following are inspired from the fundamental idea of upgrading the concepts of genetic algorithms in order to obtain a hybrid model that will allow us to interpret evolution as a whole in the following sense:

A holistic interpretation of evolution should not only be restricted to the evolutionary process of a species as done within the concept of genetic algorithms but also has to consider the evolution of the individual as well as the evolution of the welfare system and the culture in which the evolutionary process is imbedded. Applied to evolutionary computation this may be interpreted in the following way:

- Genetic algorithms, as commonly proposed in literature, model the evolution of a single species because crossover is allowed between all individuals. Natural evolution on the other hand involves the competition of a variety of species against each other in order to combat for the spreading of the usually restricted resources. A new basic model that introduces those aspects of width search within the theory of Genetic Algorithms will be presented and discussed in subsection 4.2.
- The evolution of the individual within the scope of genetic algorithms occurs in some already mentioned hybrid variants (e.g. hybrid GAs with hillclimbing) where local information about the neighborhood is used in order to improve the individual fitness of each candidate in each generation. As this is done using the same data structures as for reproduction, these 'local' improvements will be inherited to the next generation which represents a direct analogy to Lamarck's theory of evolution [5].
- The evolution of the welfare system or the culture certainly denotes the most complex part of artificial evolution:
  It somehow would have to try to model an exchange of information not only between individuals but also between subpopulations or allow populations to learn from prior populations. Some of these aspects are covered by hybrids involving genetic algorithms and tabu search. Other considerations concerning this aspect of evolution will be included in the newly introduced segregative genetic algorithms (SEGA) approach as introduced in subsection 4.1, that has primary been developed in order to avoid premature convergence.

In contrast to contributions in the field of genetic algorithms that introduce new coding standards and operators for certain problems, the introduced approach as well as the hybrids based upon this approach should be considered as heuristics 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 corresponding genetic algorithm. The additional aspects as being proposed within the scope of the present paper are inspired from optimization as well as from the views of bionics in the above mentioned sense.

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

The handling of selective pressure in the context of genetic algorithms mainly depends on the choice of a certain replacement scheme. The ‘generational replacement scheme’, for example, replaces the entire population by the next one whereas ‘elitism replacement’ keeps the best individuals of the last generation and only replaces the rest and therefore usually performs faster. On the other hand, elitism likely causes too homogeneous populations, i.e. little population diversity, and therefore might cause unwanted premature convergence. Anyway, there exists no manageable model for controllable selective pressure handling within the theory of genetic algorithms. Therefore, we introduce some kind of intermediate step (a 'virtual population') into selection which provides a handling of selective pressure very similar to that of evolution strategies. As we will exemplarily point out, the most common replacement mechanisms can easily be implemented in this intermediate selection step. Furthermore, this evolution strategy like variable selective pressure will help us to steer the degree of population diversity on the one hand and, on the other hand, it will act as a basic model for new hybrid metaheuristics based upon genetic algorithms as being proposed in section 4.
Actually, all modifications that are and will be taken into account use exactly the same operators for crossover and mutation as a corresponding genetic algorithm. As no further problem specific information is used, the new hybrids can be applied to a huge number of problems - namely at least all problems genetic algorithms can be applied to.
Similar to any other commonly used genetic algorithm we use a population of fixed size that will evolve to a new population of the same size by selection, crossover, and mutation.

What we additionally do is to introduce an intermediate step in terms of a so-called virtual population of variable size where the size of the virtual population usually has to be greater than the population size. This virtual population is created by selection, crossover, and mutation in the common sense of genetic algorithms. But like in the context of evolution strategies, only a certain percentage of this intermediate population will survive what represents a direct analogy to selective pressure handling in the context of evolution strategies.

In the terminology of evolution strategies μ parents produce λ descendants from which the best μ survive and 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 [10]). 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.

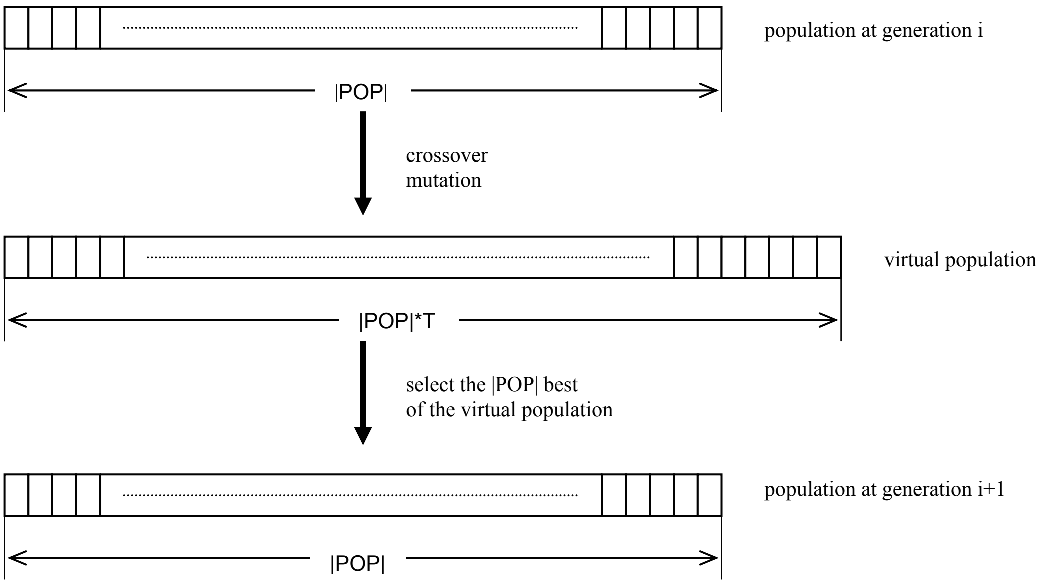

Figure 1. Evolution of a new population with selective pressure s= $\frac{1}{T}$
for a virtual population built up in the sense of a (μ, λ)-ES.

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.

Bionically interpreting this (μ, λ)-evolution strategy like selective pressure handling, for genetic algorithms this means, that some kind of 'infant mortality' has been introduced in the sense that a certain ratio of the population (|POP| * T - |POP| = |POP| * (T-1)) will never become procreative, i.e. this weaker part of a population will not get the possibility of reproduction. Decreasing this 'infant mortality', i.e. reducing the selective pressure during the evolutionary process also makes sense in a bionic interpretation because also in nature stronger and higher developed populations suffer less from infant mortality. From the point of view of optimization, decreasing the temperature during the optimization process means that a greater part of the search space is explored at the beginning of evolution - whereas at a later stage of evolution, when the average fitness is already quite high, a higher selective pressure is quite critical in that sense that it can easily cause premature convergence. Anyway, operating with a temperature converging to zero, this (μ, λ)-evolution strategy like selective pressure model for genetic algorithms acts like the corresponding genetic algorithm with generational replacement.

Moreover, implementing the analogue to the ($\mu + \lambda$)-evolution strategy denotes the other extreme of immortal individuals. However, also the implementation of this strategy is quite easy with our model as indicated in Figure 2.

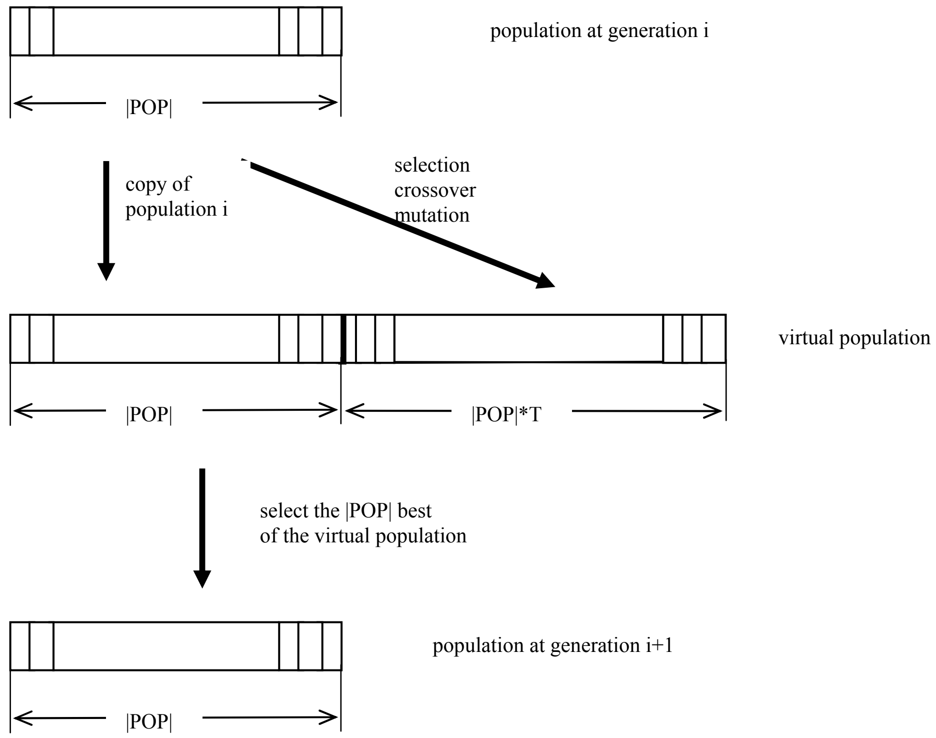

Figure 2. Evolution of a new population with a virtual population built up in the sense of a (μ+λ)-ES.

Other replacement mechanisms, like elitism or the goldcage-model for example, are also very easy to handle by just adding the best individuals respectively the best individual of the last generation to the virtual generation.

In the following section we are going to discuss new aspects and models built up upon the described variable selective pressure model. Nevertheless, there is nothing against using a genetic algorithm equipped with the variable selective pressure model without further extensions. Experiments performed on the variable selective pressure model already indicate the supremacy of this approach. In these experiments the temperature is slowly cooled down to zero. If one is aware of an appropriate measure of population diversity, the temperature can be set dynamically in order to steer the genetic diversity which seems to represent a further promising approach.

4. NEW CONCEPTS BASED UPON THE VARIABLE SELECTIVE PRESSURE MODEL

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

4.1. 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.

A critical problem in studying premature convergence is the identification of its occurrence and the characterization of its extent. Srinivas and Patnaik [11], 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. [12]) where the decrease of population diversity is the primary reason for premature convergence. However, a very homogeneous population, i.e. little population diversity, is the major reason for a genetic algorithm to prematurely converge. Therefore, this situation represents some kind of analogy to the situation when a neighborhood based search technique like simulated annealing is unable to escape from a local minimum.

The principal 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 because of too similar individuals within the subpopulations. Then a reunification from n to (n-1) subpopulations is done.

Whereas island models for genetic algorithms (e.g. in [14]) are mainly driven by the idea of using simultaneous computer systems, SEGA attempts to utilize migration more precisely in order to achieve superior results in terms of global convergence.

Figure 3 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].

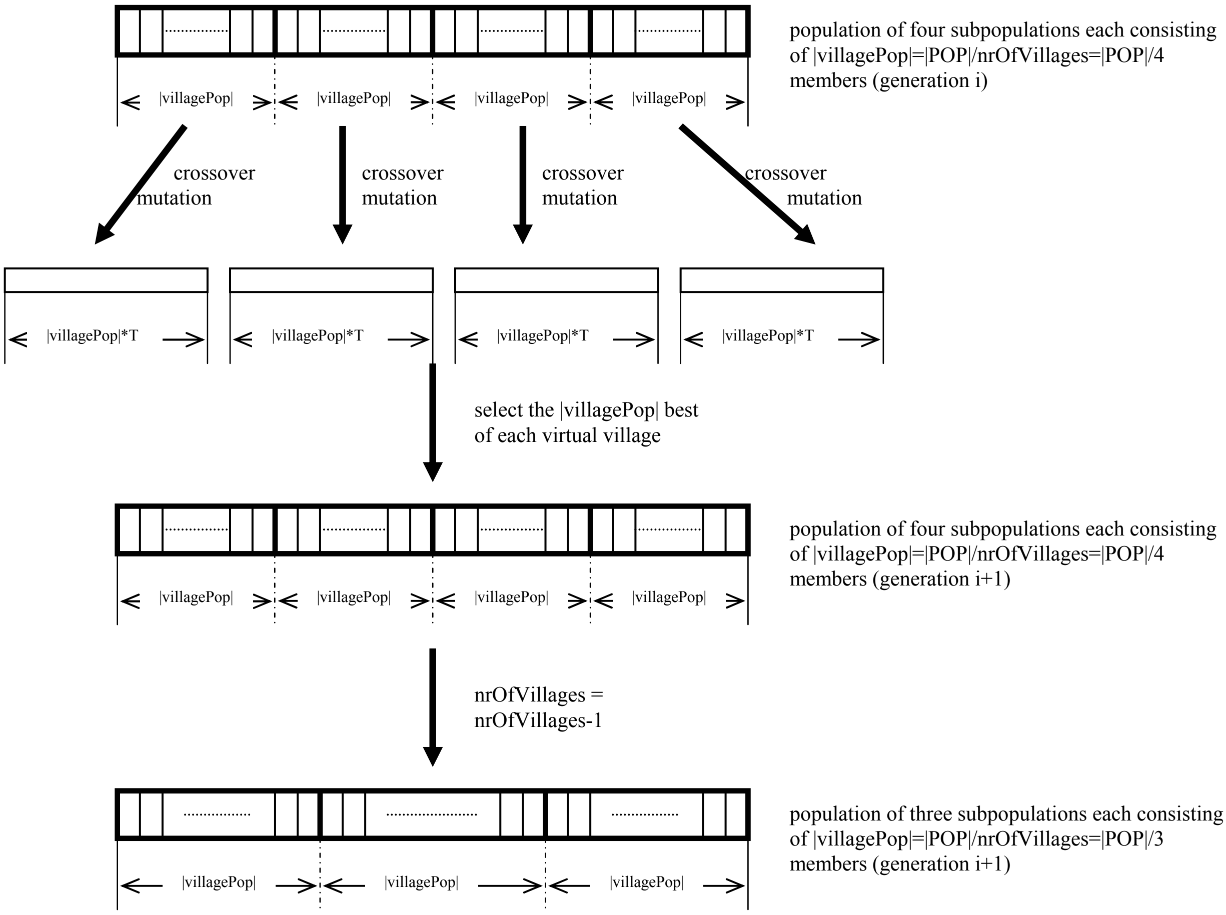

Figure 3. Evolution of a new population for the instance that four subpopulations are merged to three.

4.2. THE USAGE OF MULTIPLE CROSSOVER OPERATORS IN PARALLEL

Genetic algorithms 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 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 is necessary to choose the size of those subpopulations dynamically, 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 and mutation operators) 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 genetic algorithm known to the author 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 genetic algorithms with the certain operators one after another. Thus, the achieved profits would 'only' concern the performance of the algorithm.

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 (habitat) of each virtual subpopulation is defined by its success during the last iterations as illustrated in Figure 4.

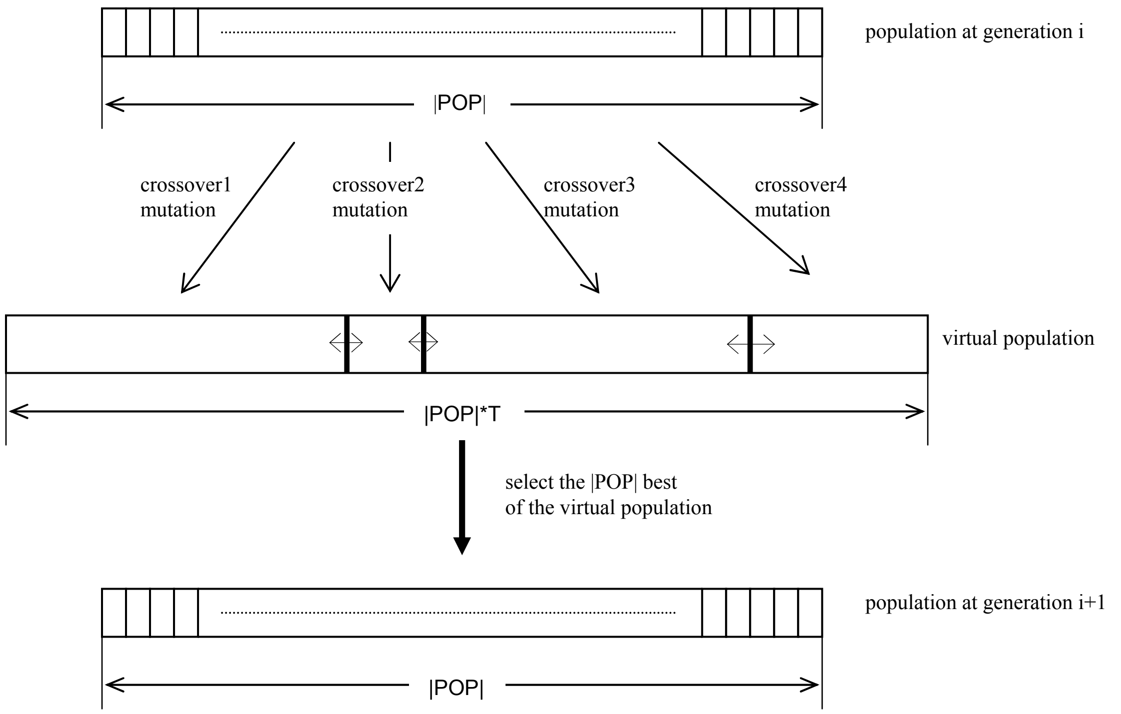

Figure 4. Evolution of a new population for the instance that four crossover operators are used in parallel.

Strictly interpreted, the term 'Multi-Creature Genetic Algorithms' might be too strong for the construction shown in Figure 4 and we might better term this extension as a Genetic Algorithm that dynamically uses multiple crossover operators in parallel.
Exemplarily considering the properties of the OX and the ERX operators for crossover it is reported (e.g. in [6]) that the OX-operator significantly outperforms the ERX-operator in terms of speed whereas the ERX-operator surpasses OX in terms of global convergence. Dynamically using multiple crossover operators in parallel utilizes the 'fast' OX-operator for a long evolution period until the performance in terms of solution quality of ERX outperforms OX at a later stage of evolution.
Anyway, this dynamic (self-organizing) strategy seems particularly suitable for problem situations where a couple of crossover operators whose properties are not exactly known are taken into account.

5. EXPERIMENTAL RESULTS

In our experiments, 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 4.1 coupled with dynamic habitat adaptation (subsection 4.2). In doing so, we have used a selection of symmetric as well as asymmetric benchmark problem instances taken from the TSPLIB [8] using updated results for the best or at least the best known solutions taken from [9]. 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 [13]. The GA and its scions use a mutation probability of 0.05 and a combination of OX-crossover [6] and ERX-crossover [6] combined with the golden-cage population model (e.g. [10]), 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 5 shows the experimental results for the problem ch130 (130 city problem) as an example of a symmetric 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.

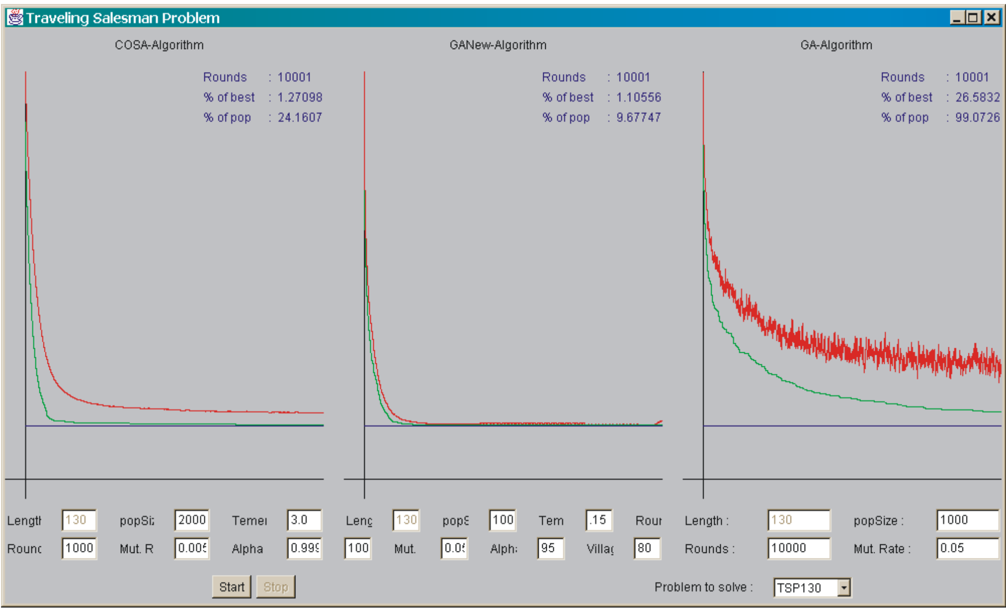

Figure 5. Comparison of COSA, SEGA with dynamic usage of more than one crossover operator, and GA on the basis of the kro124p benchmark problem: For each algorithm, the average fitness and the fitness of the best member of the population is diagrammed relatively to the best known solution represented by the horizontal line.

Table 1 shows the experimental results of COSA, GA, GA with variable selective pressure, and SEGA with dynamic habitat adaptation, 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	GA+var.Sel.Pr.	SEGA+dyn.HabitatAdapt.
Eil76 (symmetric)	5000	1.78	9.76	4.21	0.87
Ch130 (symmetric)	10000	1.27	26.58	8.91	1.10
kroA150 (symmetric)	10000	4.06	40.97	10.36	2.21
kroA200 (symmetric)	15000	5.56	45.11	14.77	2.40
br17 (asymmetric)	100	0.00	0.00	0.00	0.00
Ftv55 (asymmetric)	5000	44.22	33.92	4.02	0.54
Kro124p (asymmetric)	10000	26.78	37.49	20.06	1.22
Ftv170 (asymmetric)	15000	202.33	131.61	73.55	2.01

Table 1. Experimental results of COSA, GA, GA+variable selective pressure, and SEGA with dynamic usage of multiple crossover operators in parallel.

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 evolution 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.

Concerning the speed of SEGA, it has to be pointed out that the superior performance concerning convergence requires a higher running time, mainly because of the greater population size |POP| required. This should allow to transfer already developed GA-concepts to increasingly powerful computer systems in order to achieve better results. Using simultaneous computers seems especially suited to increase the performance of SEGA.

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.

Even though, because of better comparability, no additional hybrid techniques, like commonly used hill-climbing or certain other pre- or post-optimization techniques have been considered in the examples presented in the experimental results section, there absolutely exists no objection of doing so in order to improve the convergence.

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. Current experiments indicate that the standard deviation of the fitness distribution might be a practical generic approximation for genetic diversity.

REFERENCES

[1] AFFENZELLER M., *A New Approach to Evolutionary Computation: Segregative Genetic Algorithms (SEGA)*, IWANN'2001: Lecture Notes in Computer Science 2084, Springer Verlag, Berlin Heidelberg New York 2001, pp. 594-601.

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

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

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

[5] LAMARCK J.B., *Zoological Philosophy*, London 1914.

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

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

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

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

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

[11] SRINIVAS M. and PATNAIK L., *AdaptiveProbabilities of Crossover and Mutation in Genetic Algorithms* , IEEE Transactions on Systems, Man, and Cybernetics 24(4) 1994, pp. 656-667.

[12] SMITH S. et al., *Population Diversity in an Immune System Model: Implications for Genetic Search*, Foundations of Genetic Algorithms 2 1993, pp. 153-166.

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

[14] WHITLEY D., *A Genetic Algorithm Tutorial*, Statistics and Computing 4 1994, pp. 65-85.