

# Analysis of the Dynamics of Allele Distribution for some Selected GA-variants

Michael Affenzeller, Stefan Wagner, Stephan Winkler, Andreas Beham

School of Informatics, Communications and Media  
Upper Austria University of Applied Sciences  
Softwarepark 11, 4232 Hagenberg  
Email: michael.affenzeller@heuristiclab.com

**Abstract**—This paper exemplarily points out how essential genetic information evolves during the runs of certain selected GA-variants. The discussed algorithmic enhancements to a standard genetic algorithm certify the survival of essential genetic information by supporting the survival of relevant alleles rather than the survival of above average chromosomes. This is achieved by defining the survival probability of a new child chromosome depending on the child's fitness in comparison to the fitness values of its own parents. The described kind of analysis assumes the knowledge of the unique global optimal solution and is therefore restricted to rather theoretical considerations. The main aim of this paper is to explain the most important properties of the discussed algorithm variants in a rather intuitive way. Aspects for meaningful and practically more relevant generalizations as well as more sophisticated experimental analyses are indicated.

## I. INTRODUCTION

In this paper we try to look inside the internal functioning of several genetic algorithm (GA) variants. The discussed generic hybrids couple aspects of genetic algorithms with selection principles basically inspired by evolution strategies [1] and self adaptive measures comparable to the parameter-less GA [2]. For this purpose we use the information about globally optimal solutions which is available only for well studied benchmark problems of moderate dimension. Of course, the applied optimization strategies (i.e., in our case variants of GAs) are not allowed to use any information about the global optimum; we just use this information for analysis purposes in order to obtain a better understanding of the internal functioning and the dynamics of the discussed algorithmic concepts.

Concretely, we aim to observe the distribution of the alleles of the unique global optimal solution (denoted as essential genetic information) over the generations in order to observe the ability of the certain algorithmic variants to preserve and eventually regain essential genetic material during the run of the algorithm.

The main aim of this contribution is not to give a comprehensive analysis of many different problem instances, but

rather to highlight the main characteristics of the certain algorithm variants. For this kind of analysis we have chosen the traveling salesman problem (TSP), mainly because it is a well known and well analyzed combinatorial optimization problem and a lot of benchmark problem instances are available. We here concentrate on the *ch130* TSP instance taken from the TSPLib [3], for which the unique globally optimal tour is known; the characteristics of the global optimum of this 130 city TSP instance are exactly the 130 edges of the optimal tour which denote the essential genetic information.

In a broader interpretation of the elementary building block theory ([4], [5]) these alleles should on the one hand be available in the initial population of a GA run, and on the other hand maintained during the run of the algorithm. If essential genetic information is lost during the run, then mutation is supposed to help regaining it in order to be able to eventually find the globally optimal solution (or at least a solution which comes very close to the global optimum). In order to observe the actual situation in the population we display each of the 130 essential edges as a bar indicating the saturation of each allele in the population so there are in total 130 bars. The disappearance of a bar therefore indicates the loss of the corresponding allele in the entire population, whereas a full bar indicates that the certain allele occurs in each individual (which is the desired situation at the end of an algorithm run). As a consequence, the relative height of a bar stands for the actual penetration level of the corresponding allele in the individuals of the population and the observation of the dynamic behavior allows observing the distribution of essential genetic information during the run.

In the following, the distribution of essential genetic information and its impact on achievable solution quality will be discussed for the standard GA, a GA variant including offspring selection (OS) [6] as well as for the relevant alleles preserving GA (RAPGA) [7]. A short introduction to these algorithms is given at the beginning of the certain sections where these further developed GA-variants are tested. For a more comprehensive discussion about offspring selection and RAPGA the interested reader is referred to [8]. The results shown in the following sections are presented in a rather

The work described in this paper was done within the Josef Ressel centre for heuristic optimization sponsored by the Austrian Research Promotion Agency (FFG).

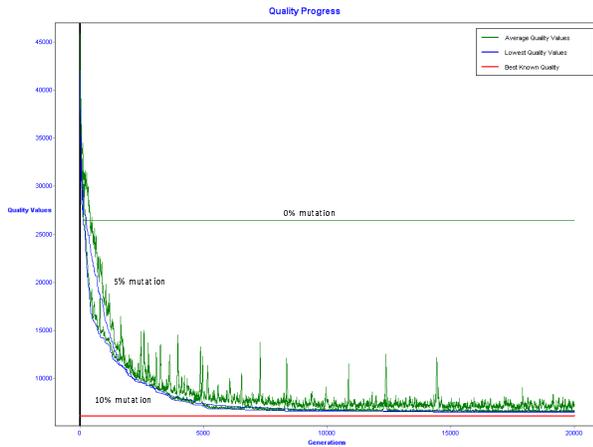


Fig. 1. Quality progress for a standard GA with OX crossover for mutation rates of 0%, 5%, and 10%.

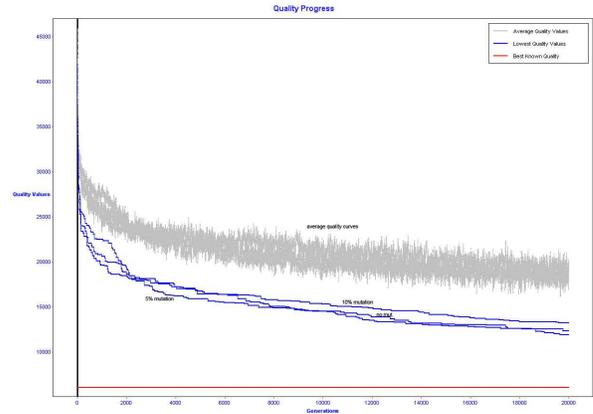


Fig. 2. Quality progress for a standard GA with ERX crossover for mutation rates of 0%, 5%, and 10%.

illustrative way and aim to provide a rather intuitive approach to the discussed algorithm variants<sup>1</sup>

## II. BUILDING BLOCK ANALYSIS FOR STANDARD GENETIC ALGORITHMS

For observing the distribution of essential alleles in a standard GA we have used the following test strategy: First, our aim was to observe the solution quality achievable with parameter settings that are quite typical for such kinds of GA applications (as given in Table I) using well known operators for the path representation, namely OX and ERX [9]; each algorithmic variant has been analyzed applying no mutation as well as mutation rates of 5% and 10%.

TABLE I  
PARAMETERS FOR TEST RUNS USING A CONVENTIONAL GA.

Parameters for the conventional GA tests (Results are graphically presented in Figures 1 and 2)	
Generations	20'000
Population Size	100
Elitism Solutions	1
Mutation Rate	0.00 or 0.05 or 0.1
Selection Operator	Roulette
Crossover Operator	OX (Fig. 1) or ERX (Fig. 2)
Mutation Operator	Simple Inversion

The following Figures 1 and 2 show the fitness curves (showing best and average solution qualities of the GA's population as well as the best known quality) for a standard GA using order crossover OX (see Figure 1) and edge recombination crossover ERX (Figure 2); the parameter settings used for these experiments are given in Table I.

For the OX crossover, which achieved the best results with the standard parameter settings, the results are shown in Figure 1; it is observable that the use of mutation rates of 5% and 10% leads to achieve quite good results (about 5% to 10%

worse than the global optimum), whereas disabling mutation leads to a rapid loss of genetic diversity so that the solution quality stagnates at a very poor level.

The use of the more edge preserving crossover operator ERX (for which results are shown in Figures 2) shows different behavior in the sense that applying the same parameter settings as used for the OX the results are rather poor independent of the mutation rate. The reason for this is just that this operator requires more selection pressure (as for example tournament selection with tournament size 3); when applying higher selection pressures it is possible to achieve comparably good results also with ERX. Still, also when applying parameter settings which give good results with appropriate mutation rates, the standard GA fails dramatically when disabling mutation<sup>2</sup>.

Summarizing these aspects we can state for the SGA applied to the TSP that several well suited crossover operators<sup>3</sup> require totally different combinations of parameter settings in order to make the SGA produce good results. Considering the results achieved with the parameter setting as stated in Table I, the use of the OX yields good results (around 10% worse than the global optimum) whereas the use of ERX leads to unacceptable results (more than 100% worse than the global optimum). On the contrary, tuning the residual parameters (population size, selection operator) for ERX would cause poor solution quality for OX.

Thus, an appropriate adjustment of selection pressures is of critical importance; as we will show in the following, self-adaptive steering of the selection pressure is able to make the algorithm more robust as selection pressure is adjusted automatically according to the actual requirements.

Figure 3 shows the distribution of the 130 essential alleles of

<sup>2</sup>In order to keep the discussion compact and on an explanatory level, detailed parameter settings and the corresponding statistically relevant result tables are not stated in this paper; detailed results for the TSP tests are given at the website of the book [8] <http://gagp2008.heuristiclab.com/material/statisticsTSP.html>

<sup>3</sup>OX and ERX are both edge preserving operators and therefore basically suited for the TSP.

<sup>1</sup>For statistically more relevant experimental results the interested reader is referred to <http://gagp2008.heuristiclab.com/material/statisticsTSP.html> where numerous result tables are provided.



Fig. 3. Distribution of the alleles of the global optimal solution over the run of a standard GA using OX crossover and a mutation rate of 5% (remaining parameters are set according to Table I).

the unique globally optimal solution over time for the overall best parameter constellation found in this section, i.e. the use of OX crossover with 5% mutation rate. In order to make the snapshots for the essential allele distribution within the SGA’s population comparable to those captured applying a GA with offspring selection or the RAPGA, the timestamps are not given in iterations but in the number of evaluations (which is in the case of the SGA equal to the population size times the number of generations executed).

Until after about 10.000 evaluations, i.e. at generation 100, we can observe quite typical behavior, namely the rise of certain bars (representing the existence of edges of the global optimum). However, what happens between the 10.000th and 20.000th evaluation is that some of the essential alleles (about 15 in our test run) become fixed whereas the rest (here about  $130 - 15 = 115$  in our test run) disappears in the entire population. As we can see in Figure 1, without mutation the genetic search process would already be over at that moment due to the fixation of all alleles; from now on mutation is the driving force behind the search process of the SGA. Most of the essential alleles can be reintroduced and fixed approximately between the 20.000th and 2.000.000th evaluation.

However, even if this procedure is able to fulfill the function of optimization reasonably good when applying adjusted parameters, it has not much in common with the desired functioning of a genetic algorithm as stated in the schema theorem and the according building block hypothesis. According to this theory, we expect a GA to systematically collect the essential pieces of genetic information which are initially spread over the chromosomes of the initial population as reported for the canonical GA. As we will point out in the next sections, GAs with offspring selection as well as RAPGA are able to considerably support a GA to function in exactly that way even under not so idealized conditions as required in the context of the canonical GA.

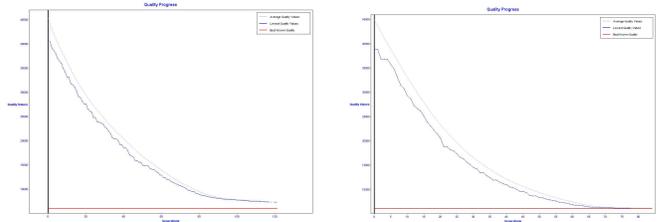


Fig. 4. Quality progress for a GA with offspring selection and OX (left figure), ERX (right figure) and a mutation rate of 5%. (remaining parameters are set according to Table II)

#### A. Building Block Analysis for GAs Using Offspring Selection

When using offspring selection (OS), newly generated solution candidates are selected as individuals of the next generations’s population if they meet some given criterion. In its strict variant this means that children are discarded unless their fitness value is better than the fitness value of the better of the two parents and a new generation is filled up only with successful offspring. In each generation, selection pressure  $selPres$  is in this context calculated as the ration of the number of produced offspring and the number of successful offspring that meet the given success criterion:  $selPres = \frac{[generated\ solutions]}{[successful\ solutions]}$ , with a maximum value of selection pressure acting as a quite intuitive and meaningful termination criterion.

The aim of this section is to highlight some characteristics of the effects of offspring selection. As the termination criterion of a GA with offspring selection is self-triggered, the effort of these test runs is not constant; however, the parameters given in Table II are adjusted in a way that the total effort is comparable to the effort of the test runs for the SGA building block analyses discussed in Section II.

TABLE II  
PARAMETERS FOR TEST RUNS USING A GA WITH OS.

Parameter settings for the offspring selection GA runs (Results are graphically presented in Figures 4, and 5)	
Population Size	500
Elitism Solutions	1
Mutation Rate	0.00 or 0.05
Selection Operator	Random
Crossover Operator	OX , ERX, or comb. of OX, ERX, and MPX
Mutation Operator	Simple Inversion
Maximum Selection Pressure	200

Similarly as for the SGA, we here take a look at the performance of some basically suited (edge preserving) operators. The results shown in Figure 4 highlight the benefits of self-adaptive selection pressure steering introduced by offspring selection: Independent of the other parameter settings, the use of all considered crossover operators yields results near the global optimum.

As documented in the previous section we have observed that the standard GA heavily relies on mutation, when the selection pressure is adjusted at a level that allows the SGA to search in a goal oriented way. Therefore, we are now especially interested in how offspring selection can handle the situation when mutation is disabled. The left part of Figure 5 shows the quality curves for the use of the ERX crossover operator

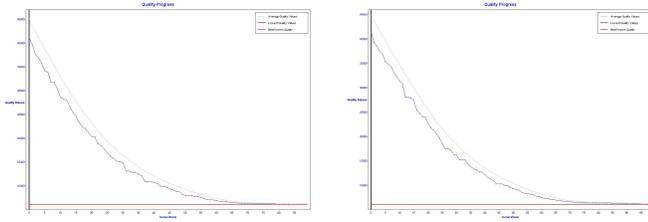


Fig. 5. Quality progress for a GA with offspring selection, ERX and no mutation (left figure) as well as the quality progress for a GA with offspring selection using a combination of OX, ERX and MPX and a mutation rate of 5% (right figure) (remaining parameters are set according to Table II).

(which achieved the best results with 5% mutation) without mutation and the same settings for the remaining parameters. The remarkable result is that the result is practically as good as with mutation, which at the same time means that offspring selection does not rely on the genetic diversity regaining aspect of mutation. Furthermore, this also means that offspring selection is able to keep the essential genetic information which in the concrete example is given by the alleles of the globally optimal solution. Offspring selection enhance the evolutionary process in such a way only that offspring of two parents take part on the ongoing evolutionary search that were successful recombinations. Therefore, in contrast to the SGA the algorithm is not only able to keep the essential genetic information, but slowly merges the essential building blocks step by step which complies with the core statements of the building block theory and is not restricted to binary encoding or the use of certain operators.

In previous publications we have even gone one step further: We have been able to show in [10] that even with crossover operators basically considered unsuitable for the TSP (as they inherit the position information rather than the edge information like CX or PMX [9]) it becomes possible to achieve high quality results in combination with offspring selection. The reason is the sufficiency that a crossover operator is able to produce good recombinations from time to time (as only these are considered for the future gene pool); the price which has to be paid is that higher average selection pressure has to be applied, if the crossover operator is more unlikely to produce successful offspring.

As a proof of concept for applying more than one crossover at the same time, we have repeated the previous test runs with OX, MPX and ERX with the only difference that for these tests all crossover operators have been used. The right part of Figure 5 shows the quality curves (best and average results) for this test run and shows that the results are in the region of the global optimal solution and therefore at least as good as in the test runs before. A further question that comes along using multiple operators at once is their performance over time: Is the performance of each of the certain operators relatively constant over the run of the algorithm?

In order to answer this question, Figure 6 shows the ratio of successful offspring for each crossover operator used (in the sense of strict offspring selection which requires that

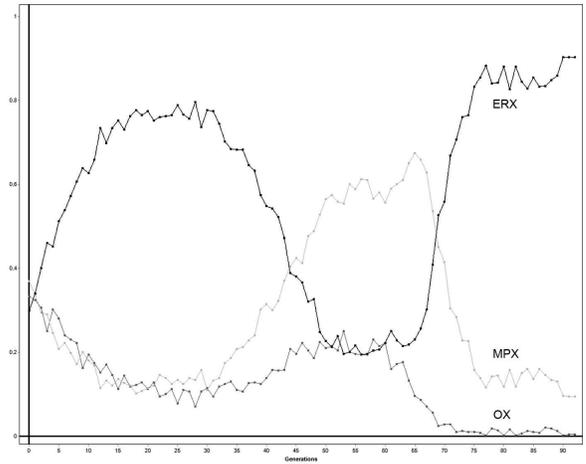


Fig. 6. Success progress of the different crossover operators OX, ERX and MPX and a mutation rate of 5%. The plotted graphs represent the ratio of successfully produced children to the population size over the generations.

successful children have to be better than both parents). Figure 6 shows that ERX performs very well at the beginning (approximately until generation 45) as well as in the last phase of the run (circa from gen. 75). In between (approximately from generation 45 to generation 75), when the contribution of ERX is rather low, MPX shows significantly better performance. The performance of OX in terms of its ability to generate successful offspring is rather mediocre during the whole run showing very little success in the last phase. The analysis of reasons of the behavior of the certain operators over time would be an interesting field of research; anyway, it is already very interesting to observe that the performance characteristics of the operators are changing over time to such an extent.

For a more detailed observation of the essential alleles during the runs of the GA using offspring selection we show the allele distribution for the ERX crossover, which achieved slightly better results than the other crossover operators, in Figure 7. However, the characteristics of the distribution of essential alleles are quite similar also for the other crossover operators when using offspring selection. As a major difference in comparison to the essential allele distributions during a standard GA, we can observe that the diffusion of the essential alleles is established in a rather slow and smooth manner. The essential alleles are neither lost nor fixed in the earlier stages of the algorithm, so the bars indicating the occurrence of the certain essential alleles (edges of the optimal TSP path) in the entire population are growing steadily until almost all of them are fixed by the end of the run.

This behavior not only indicates a behavior in accordance with the building block hypothesis, but also implies that the algorithm performance no more relies on mutation to an extent as observed for the corresponding SGA analyses. In order to confirm this assumption we have repeated the same test without mutation and indeed, as it can be seen by a comparison of the two pictures shown in Figure 7, the saturation behavior of the essential building blocks is basically the same, no

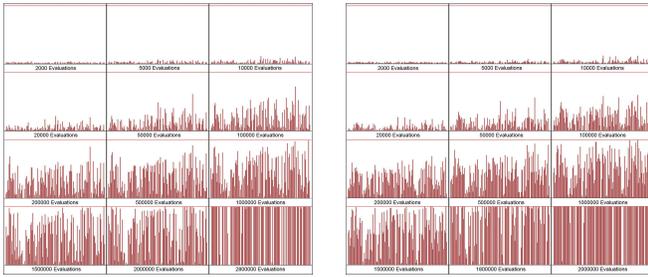


Fig. 7. Distribution of the alleles of the global optimal solution over the run of an offspring selection GA using ERX crossover and a mutation rate of 5% (left picture) and no mutation (right picture) (remaining parameters are set according to Table II).

matter if mutation is used or not. This is a remarkable observation as it shows that offspring selection enables a GA to collect the essential building blocks represented in the initial population and compile high quality solutions very robustly in terms of parameters and operators like mutation, selection pressure, crossover operators etc. This property is especially important when exploring new fields of application where suitable parameters and operators are usually not known a priori.

### B. Building Block Analysis for the Relevant Alleles Preserving GA (RAPGA)

Similar to the previous section we aim to highlight some of the most characteristic features of the relevant alleles preserving GA (RAPGA). The RAPGA ideally works in such a way that new child solutions are added to the new population as long as it is possible to generate unique and successful offspring stemming from the gene pool of the last generation.

This idea is implemented using ad hoc population size adjustment in that sense that potential offspring generated by the basic genetic operators are accepted as members of the next generation if and only if they are able to outperform the fitness of their own parents and if they are new in that sense that their chromosome consists of a concrete allele alignment that is not represented yet in an individual of the next generation. As long as new and (with respect to the definition given in the context of OS) successful individuals can be created from the gene pool of the actual generation, the population size is allowed to grow up to a maximum size. Similar to OS, a potential offspring which is not able to fulfill these requirements is simply not considered for the gene pool of the next generation.

Still a lower as well as an upper limit of population size are necessary in order to achieve efficient algorithmic performance. In order to terminate the run of a certain generation in case it is not possible to fill up the maximally allowed population size with new successful individuals, an upper limit of effort in terms of generated individuals is also necessary. This maximum effort per generation is the maximum number of newly generated chromosomes per generation (no matter if these have been accepted or not). In order to terminate the run of a certain generation in case it is not possible to fill up the maximally allowed population size with new successful

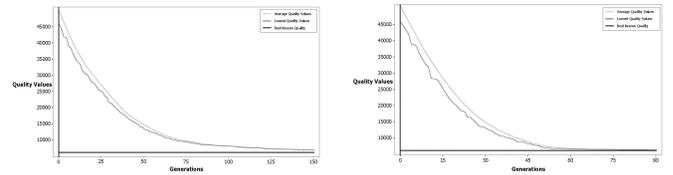


Fig. 8. Quality progress for a relevant alleles preserving GA with OX (left figure) and ERX (right figure); both with a mutation rate of 5%.

individuals, an upper limit of effort in terms of generated individuals is necessary. This maximum effort per generation is the maximum number of newly generated chromosomes per generation (no matter if these have been accepted or not). The question, whether or not an offspring is better than its parents, is answered in the same way as in the context of offspring selection.

TABLE III  
PARAMETERS FOR TEST RUNS USING THE RELEVANT ALLELES PRESERVING GENETIC ALGORITHM.

Parameters for the RAPGA tests (Results presented in Fig.8 and Fig.9)	
Max. Generations	1'000
Initial Population Size	500
Mutation Rate	0.00 or 0.05
Elitism Rate	1
Male Selection	Roulette
Female Selection	Random
Crossover Operator	OX ERX combined (OX, ERX and MPX)
Mutation Operator	Simple Inversion
Minimum Population Size	5
Maximum Population Size	700
Effort	20'000
Attenuation	0

The following experiments are set up quite similar to the offspring selection experiments of the previous section. Firstly, the considered operators OX, and ERX as well as the combination (OX, MPX, ERX) are applied to the *ch130* benchmark TSP problem taken from the TSPLib. Then the most successful operator or operator combination, respectively, is also exemplarily considered without mutation in order to show that the RAPGA like offspring selection does not rely on mutation to such an extent as conventional GAs.

Already the experiments using OX (see Figure 8) show good results (approximately 5% – 10% worse than the globally optimal solution) which are even slightly better than the corresponding offspring selection results. Even if only single test runs are shown in this paper it has to be pointed out that the authors have taken care that characteristic runs are shown. Besides, as described in the more systematical experiments described in [8], especially due to the increased robustness caused by offspring selection and RAPGA the variance of the results' qualities is quite small.

Similar to what we stated for the OS analyses, also for the RAPGA the best results could be achieved using ERX (a comparison between OX and ERX is shown in Figure 8) as well as using the combination of OX, ERX and MPX (left part of Figure 9). The achieved results using these these operators are about 1% or even less worse than the global optimal solution. In the case of the RAPGA the operator combination

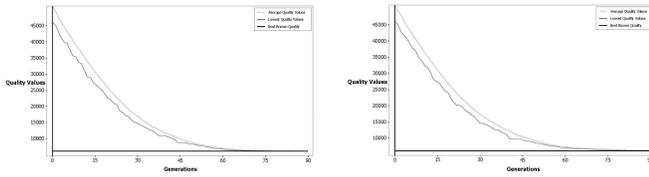


Fig. 9. Fitness curve for a relevant alleles preserving GA using a combination of OX, ERX and MPX and a mutation rate of 5% (left figure) and mutation switched off (right figure).

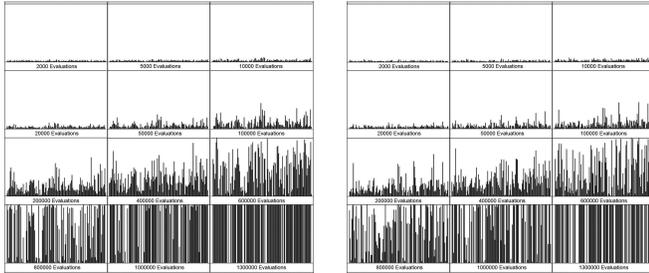


Fig. 10. Distribution of the alleles of the global optimal solution over the run of a relevant alleles preserving GA using a combination of OX, ERX and MPX and a mutation rate of 5% (left figure) and mutation switched off (right figure) (remaining parameters are set according to Table III).

turned out to be slightly better than ERX (in 18 of 20 test runs). Therefore, this is the operator combination we have also considered for a detailed building block analysis without mutation as well as applying 5% mutation.

Barely surprising, the results of RAPGA with the operator combination consisting of OX, ERX, and MPX turned out to be quite similar to those achieved using offspring selection and the ERX operator. Due to the name giving aspect of essential allele preservation, disabling mutation (see Figure 9) has almost no consequences concerning achievable global solution quality. Even without mutation the results are just 1-2% worse than the global optimum. The distributions of essential alleles over the generations of the RAPGA run (as shown in Figure 10) also show quite similar behavior as already observed in the corresponding analyses of the effects of offspring selection. Almost all essential alleles are represented in the first populations and their diffusion is slowly growing over the GA run, and even without mutation the vast majority of essential alleles is fixed by the end of the RAPGA runs.

Summarizing the test results, we can state that quite similar convergence behavior is observed for a GA with offspring selection and the RAPGA, which is characterized by efficient maintenance of essential genetic information. As shown in Section II, this behavior (which we would intuitively expect from any GA) cannot be guaranteed in general for GA applications where it was mainly mutation which helped to find acceptable solution qualities.

### III. CONCLUSIONS

In this paper we have exemplarily opposed the characteristic behavior of conventional GA with the typical functioning

of generic hybrids based upon self adaptive selection pressure steering. The shown experiments are not comprehensive at all as only snapshot results are shown under somehow idealized conditions which use the knowledge of the known unique optimal solution; nevertheless these tests represent a compressed summary of the algorithm's properties which are discussed in depth in [8]. In [8] it is also shown that the observed behavior is not restricted to the TSP, but can be also observed also in other combinatorial optimization problems like the CVRP(TW) and especially in genetic programming applications. Extensive empirical studies for the TSP, which are also relevant in view of the results of this paper, are provided at the website<sup>4</sup> of the book where also tests for the CVRP(TW) and genetic programming applications are documented.

### REFERENCES

- [1] H. G. Beyer, *The Theory of Evolution Strategies*. Springer, 2001.
- [2] F. G. Lobo and D. Goldberg, "The parameter-less genetic algorithm in practice," *Information Sciences*, vol. 167, no. 1-4, pp. 217-232, 2004.
- [3] G. Reinelt, "TSPLIB - A traveling salesman problem library," *ORSA Journal on Computing*, vol. 3, pp. 376-384, 1991.
- [4] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison Wesley Longman, 1989.
- [5] C. R. Stephens and H. Waelbroeck, "Schemata evolution and building blocks," *Evolutionary Computation*, vol. 7, no. 2, pp. 109-124, 1999.
- [6] M. Affenzeller, S. Wagner, and S. Winkler, "Goal-oriented preservation of essential genetic information by offspring selection," in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO) 2005*, vol. 2. Association for Computing Machinery (ACM), 2005, pp. 1595-1596.
- [7] —, "Self-adaptive population size adjustment for genetic algorithms," in *Proceedings of Computer Aided Systems Theory: EuroCAST 2007*, ser. Lecture Notes in Computer Science, R. M. Diaz, F. Pichler, and A. Quesada, Eds. Springer, 2007, pp. 820-828.
- [8] M. Affenzeller, S. Winkler, S. Wagner, and A. Beham, *Genetic Algorithms and Genetic Programming: Modern Concepts and Practical Applications*. CRC Press, 2009.
- [9] P. Larranaga, C. M. H. Kuijpers, R. H. Murga, I. Inza, and D. Dizdarevic, "Genetic algorithms for the travelling salesman problem: A review of representations and operators," *Artificial Intelligence Review*, vol. 13, pp. 129-170, 1999.
- [10] M. Affenzeller and S. Wagner, "SASEGASA: A new generic parallel evolutionary algorithm for achieving highest quality results," *Journal of Heuristics - Special Issue on New Advances on Parallel Meta-Heuristics for Complex Problems*, vol. 10, pp. 239-263, 2004.

<sup>4</sup> <http://gagp2008.heuristiclab.com/material.html>