Adaptive Genetic Algorithms: A new Approach for solving nonstandard Vehicle Routing Problems efficiently

Ulrich Derigs, Martin Kabath, Markus Zils

University of Cologne, WINFORS

1 Introduction

Vehicle routing is a vital area within Operations Research, appealing for researchers as well as practitioners. Yet it becomes apparent that the intention and the focus which is directing the development of solution procedures is different. The focus and main interest of researchers is devoted to testing different OR-paradigms and to apply theoretical or methodological concepts to the development of algorithms, which are able to produce high quality solutions in short running times. Therefore researchers concentrate their effort on the discussion of several standard problems for which sets of demanding benchmark instances have been established. In contrast to this practitioners as owner of specific, in general so far unstructured and not yet formalized problem types or even single instances are requiring systems to be designed and implemented in short development times producing constantly satisfying solutions for different instances in acceptable turn around time. That is we can observe a mismatch of the respective goals: designing fast algorithms vs. rapid development of effective systems.
For unstructured decision problems leading to complex combinatorial optimization problems two approaches are common

- the development of a generic model and special purpose (heuristic) method as well as its coding or
- the application of a so-called metaheuristic i.e. a general purpose method as for instance Tabu-neighborhood search or Genetic Algorithms (GA)

In the second approach only an adaquate problem representation has to be identified and submitted to the metaheuristic workbench. Yet, to produce good results in a second phase time consuming and tedious calibration of parameters has to be performed.
In the following we introduce AGA - Adaptive Genetic Algorithms - an approach which eliminates the disadvantage of time consuming trial and error calibration by performing on-line calibration dynamically within an adaptive solution process. With this contribution we want to show that through this conceptual extension the widely used concept of GA can be improved to produce a general purpose heuristic which can be applied to so far unstructured managerial problems in the transportation industry, i.e. it enables practitioners to rapidly develop effective application

systems. We will briefly report results on a nonstandard vehicle routing problem. For more details on the approach in general and its application to other combinatorial optimization problems we refer to [1].

2 The concept of Adaptive Genetic Algorithms

Ideally the GA-modeler should only design a coding scheme, specify a set of potential operators together with a range and discretization of their application rates and after the completion of this creative part of the development submit the tedious and tiring job of calibration to some kind of assistant, who operates using special procedural knowledge on how to produce high-quality solutions fast. Our idea of Adaptive Genetic Algorithms (AGA) is to automatically and dynamically perform an autoconfiguration of GA-parameters which are considered to have the highest impact on solution quality: crossover-, mutation- and selection-operator.
Within AGA not only information on the solution itself is represented in the chromosomes but also information on the parameterization, the so-called environment, which was applied in the generation of this chromosome, is coded and submitted to the competition process. Thus following the GA paradigm successful parameter combinations, i.e. those which produce high quality individuals, will receive a higher probability of being inherited, so that at the end of the adaptation process high-quality parameter-combinations are expected to predominate resulting consistently in high-quality solutions.
The central problem of such an approach is to establish a link between the quality of parameter combinations and individual chromosomes. In the AGA concept this is done by defining the quality of the parameter setting by its ability to produce offsprings with better fitness values compared to other combinations of parameters. Based on this simple idea it is possible to define a scheme for dynamic autoconfiguration.
As the classical GA the new AGA is a metaheuristic, i.e. its principle can be described independently of any special problem class. Thus, for the following we assume an arbitrary problem class for which a coding scheme together with a "toolbox" of applicable GA-operators is already available. By C we denote the set of applicable crossover operators, by M the set of appropriate mutation operators and by S the set of selection operators implemented in our "toolbox".
To measure the effectiveness of a specific operator $j \in C \cup M \cup S$ we introduce a so-called scoreboard $B(j)$ which serves as an account, i.e. if the parameter proofs to be effective it will accumulate positive votings during the adaptation process, else it will be penalized by subtracting points from the scoreboard. At the beginning all operators have an identical score of 0.
With every solution/individual i in a population P we associate its so-called environment $E(i) = (c(i), m(i), s(i))$ with $c(i) \in C$, $m(i) \in M$ and $s(i) \in S$. For the initial population the operators defining an environment E are independently

generated at random and then randomly assigned to an individual. Due to the fact that the population will usually be much smaller in size than the number of potential operator combinations, not all combinations can be assigned to the initial population. But in principle all possible operator combinations can be generated during the course of the AGA due to the fact that the propagation of operators is determined separately for each of the three operator classes.

The AGA is based on a special population structure usually used in parallel implementations of GA, the so called overlapping population structure. The neighborhoods are all overlapping and pairwise different. All pairs of neighborhoods are connected by a path not longer than the number of individuals in the populations.

In the following we describe the voting-scheme and the propagation strategy of the environmental settings in detail: After the scoreboard B, the population P and the parameter environment E are initialized the simulation of the natural adaptation by reproduction is performed as long as a termination criterion is not fulfilled. The concept of parallel processing in the overlapping population structure is simulated in our AGA-scheme by selecting randomly one individual at a time as a master individual for sexual reproduction with one of its neighbors. First the master individual k is randomly selected from the population. Based on the selection operator of its environment the slave individual l is chosen from the neighborhood of k. Two offsprings are generated applying the master's reproduction operators. One individual is assigned to be the master-offspring k' and the other the slave-offspring l' with both receiving the respective environment. To identify the quality of the solutions both are evaluated with respect to the fitness function.

The master k is replaced by k' if the offspring has a better fitness value. If this replacement occurs an update of the scoreboard B is performed by increasing the account for the operators of the master individual k by one. If the slave individual l is replaced by its offspring the parameters of its environment $E(l)$ are downgraded by one in the scoreboard B, because these parameters did not take part in the actual successful reproduction, but had the chance to accumulate credit on the scoreboard in previous iterations (memory function).

If a replacement of the slave individual l by its offspring l' occurs the algorithm computes the differences in the scoreboard B for all parameter pairs of k and l. If a difference exceeds a given threshold τ, the so called *acceptance rate*, the superior parameters of the master individual k are copied to the environment of the new individual l. Since the inheritance of the operators is decided separately for the different operator categories, new parameter combinations can be generated in this step.

This procedure is repeated until a termination criterion applies. The idea of the *survival of the fittest* implies that at the end of this adaptation process the superior operators have propagated through the environment E of the population P while the inferior combinations have died out.

Table 1: SACP characteristics

Characteristic	SACP9	SACP33
# airports	9	33
# aircrafts	2 x DC8, 2 x 747	40 x 747
# Passenger Flights	8	68
# O&D-Markets	26	675
geographic scope	3 continents	global

3 Application to the Strategic AirCargo Problem

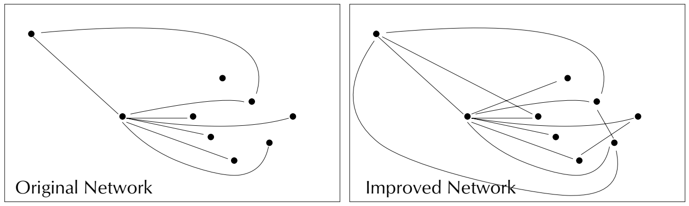
This real world problem stems from a leading internationally operating mixed cargo and passenger carrier. To fulfill the market demand given as a set of origin and destination units (O&D) a subset of flight legs from a potential network of operational feasible flight legs has to be identified. Under the constraint that the entire O&D demand can be satisfied and that the selected legs can be connected to feasible aircraft routings (so called rotations) for a given fleet of cargo aircrafts the objective function is to minimize network costs [2]. This so called Strategic AirCargo Problem (SACP) can be decomposed into two separate problems: the schedule construction and the schedule evaluation problem. The schedule construction problem consists of determining on what days in the week which aircraft should fly which route at what time. This problem is restricted by a number of operational constraints, like minimum ground times for loading and refueling, payload-range constraints etc. The schedule evaluation problem is to find feasible freight routings for the cargo on the capacitated time-space-network generated by the schedule generation process and available belly-capacity on passenger flights.

The need for the solution of such problems arises in the airline's planning departments regularly when new network structures or fleet decisions are investigated or when a one-time special request like the evaluation of potential benefits of strategic alliances due to the integration of schedules of partner airlines have to be analyzed. In both cases, incrementally changing existing schedules is not a feasible approach. Schedules have to be built from scratch, which is a non-trivial task and can consume weeks of manual planning.

In a feasibility study to support this planning task with computer based algorithms the AGA-concept was applied. According to the decomposition of the problem structure the solution approach was also divided into a schedule construction phase applying AGA and a simple heuristic based on a greedy strategy for the schedule evaluation phase.

As a coding scheme t permutations of length n, with t being the number of days in the planning horizon ($t = 7$) and n the number of stations to be served, with two entries per position (one station key and one aircraft key) are concatenated to one chromosome. Each permutation encodes a sequence of stations as indexed in

Figure 1: SACP9 Solution



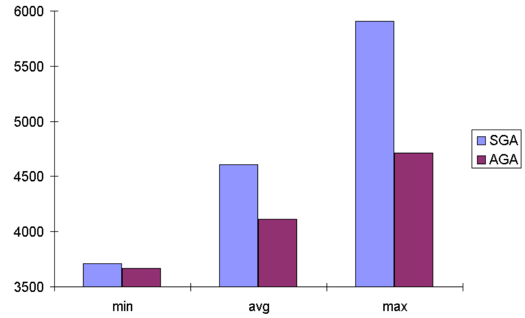
the station key. The aircraft key is a boolean variable for each individual aircraft indicating whether this station can be part of this aircrafts route. The aircraft key itself is encoded using an explicit coding table to allow multiple interpretation of an airport code in the permutation. In the decoding phase the permutations are scanned from left to right for each aircraft and a deterministic constructive procedure generates feasible rotations for all aircrafts.

In table 1 we display the characteristics of two real world instances for the two planning scenarios. Concerning the quantitative results of the SACP9 the AGA-approach was able to reduce the network costs by 5% compared to the company's original network structure. In addition to that one aircraft could be saved and utilized for other purposes increasing overall productivity and the average load factor of the remaining aircrafts. As a qualitative result AGA consistently proposed a new network structure leading to a change in the analyst's Hub & Spoke planning paradigm by introducing more direct flights and collecting cargo on round trips before transferring it to the main hub as displayed in figure 1.

Since in the SACP33-instance a future scenario for a strategic alliance was simulated no comparison to actual networks is available. The generated schedule proposes the installation of hubs different from the hubs being operated today by the individual carriers and saved two wide-body aircrafts of type Boeing 747 worth 100 Mio. US$ each. Since this strategic problem is still under investigation no final results can be published at this point.

In order to verify that a decrease in time for calibration by applying the AGA did not result in the deterioration of the solution quality possible with GA's we performed extensive tests on the small SACP9-instance: We compared the performance of a carefully selected and competitive GA (further referred to as SGA) and the AGA by explicitly probing all 630 potential parameter combinations that could be generated by the AGA from 9 crossover, 7 mutation and 10 selection operators with the efficient SGA. The AGA always outperformed the SGA concerning the overall best, the average best and the worst best solutions over all runs (see figure 2, all runs were repeated 5 times). In addition to that the AGA proofed to be robust against the variation of the threshold τ and the number of neighbors in the population topology. Also with respect to computational effort the AGA clearly outperformed the GA by only requiring 1.9% of the total number of evaluations of GA, which translated in clock time is one day vs. three month.

Figure 2: SGA vs. AGA, SACP9



4 Conclusions

The great appeal of GA is the ease of implementation and the wide range of problem classes that can be solved if only a coding-scheme can be developed. The goal of this study was to increase the ease and acceptance of applying GA-concepts in practice by eliminating one of the greatest drawbacks of the application of GA, the calibration problem. We introduced a method for dynamic autoconfiguration of those parameters with the greatest interaction and impact on the solution quality by presenting the concept of Adaptive Genetic Algorithms (AGA).

Originally AGA was designed to reduce the time for calibration and to achieve a higher degree of generalizeability. Yet, for several different problem classes we could empirically show that the AGA and the dynamic autoconfiguration concept even outperforms the classical GA consistently with respect to robustness and quality of the solutions [1]. Also our experience while developing implementations for different problem classes shows that a rapid and efficient implementation is possible, since the same *tool-box* could be applied to different combinatorial optimization problems successfully. The results on the SACP give strong evidence that results of high impact for unstructured real world managerial problems can be obtained with reasonable computational effort and in a short development time.

Our research on the application of AGA to the problems studied in this report is still far from being complete. We plan to investigate other implementations of the AGA using different coding schemes. Research on applications to other combinatorial optimization problem is under way and showing promising results.

References

[1] Ulrich Derigs, Martin Kabath, and Markus Zils. Adaptive genetic algorithms: A methodology for dynamic autoconfiguration of genetic search algorithms. WINFORS working paper submitted to Proceedings of the Metheuristic International Conference MIC'97 in Sophia Antipolis, July 1997.

[2] Markus Zils. Genetische Algorithmen zur strategischen Flotteneinsatzplanung in der Luftfrachtindustrie. Master's thesis, WINFORS, University of Cologne, 1996.