A hybrid ACO-GA approach to solve Vehicle Routing Problems

Marc Reimann[1], **Serhiy Shtovba**[2], **Erivelton Nepomuceno**[3]

[1] University of Vienna, Vienna, Austria *E-Mail:* marc.reimann@univie.ac.at

[2] Vinnitsa State Technical University, Vinnitsa, Ukraine

[3] Federal University of Minas Gerais, Belo Horizonte, Brazil

I. Introduction

In this paper we propose a hybrid approach for solving vehicle routing problems. The main idea is to combine Ant Colony Optimization (ACO) and Genetic Algorithms (GAs) to exploit (a) the strenghts of both approaches and (b) possible gains through the use of a shared memory.

The Vehicle Routing Problem (VRP) is a well known combinatorial optimization problem of large economic importance. In particular, the European situation reflects the need for improved efficiency in freight transportation, as the traffic volume increases much faster than the street network grows. Thus, given the current efficiency, this will eventually lead to a breakdown of the system. However, with rapidly increasing computational power intelligent optimization methods can be developed and used to increase the efficiency in freight transportation and circumvent the above mentioned problem.

The VRP involves the design of a set of minimum cost delivery routes, originating and terminating at a depot, which services a set of customers. Each customer must be supplied exactly once by one vehicle route. The total demand of any route must not exceed the vehicle capacity. The total length of any route must not exceed a pre-specified bound. This problem is known to be NP-hard (c.f. Garey and Johnson (1979), such that exact methods like Dynamic Programming or Branch & Bound work only for small problems in reasonable time. Thus, a large number of approximation methods have been proposed. Most of the recent approaches are based on meta-heuristics like Tabu Search, Simulated Annealing, Neural Nets, Genetic Algorithms and Ant Colony Optimization. However, most of this work is on the exclusive use of any of these techniques. While this may lead to good results, in some cases even these methods fail to find solutions close to the global optimum.

Thus, hybrid approaches could be a promising alternative. Work on Tabu Search featuring some aspects of Genetic Algorithms already proved to be very efficient (e.g. Rochat and Taillard (1995)).

As stated above, in this paper we report on a combination of Ant Colony Optimization with Genetic Algorithms. The motivation for this approach comes from two observations.

First, multi-colony(population) approaches have proven to be more powerful than their single population counterparts for both GAs and ACO. In particular, Doerner et al.(2001a) and Gambardella (1999) have proposed multi-colony ant approaches that find excellent results for different vehicle routing problems.

The second observation relates to succesfull ACO approaches. In order to find competitive solutions it is in general necessary to use some kind of post-optimization (local search) to improve the individual ants' solutions. In the proposed approach, this could be done by the GA.

Finally, Hansen and Mladenovic (1999) have recently proposed a new method called Variable Neighborhood Search, which exploits different types of neighborhoods over time to avoid getting stuck in local optima. This approach has shown to be very powerful for the Traveling Salesperson Problem (TSP). Using different strategies for different populations in evolutionary algorithms might also help to ensure that different patterns of solutions obtainable using different techniques can be combined to form even better solutions. This was confirmed by work done on a real-world vehicle routing problem (Doerner et al. (2001b)).

II. Solution method

In this section we will first describe the two approaches, Ant Colony Optimization and Genetic Algorithms independently. After that, we will propose the framework for our hybrid approach.

A. Ant Colony Optimization

The Ant Colony Optimization approach is based on the behavior of real ants searching for food. Real ants communicate with each other using an aromatic essence called pheromone, which they leave on the paths they traverse. If ants smell pheromone in their vicinity, they are likely to follow that pheromone, thus reinforcing this path. The pheromone trails reflect the 'memory' of the ant population.

In artificial terms, the optimization methods uses this intuition in the following way. Ants construct solutions by making a number of decisions probabilistically. In the beginning there is no collective memory, and the ants can only follow some local private information. As some ants have constructed solutions, pheromone information is built. This pheromone information guides other ants in their decision making. The idea is that good paths are shorter, thus ants can travel these paths faster and there will be more pheromone on these paths. However, over time pheromone on trails that are not reinforced will evaporate. Thus, over time, the shortest paths will be reflected in the pheromone information.

The ACO algorithm mainly consists of the iteration of two steps:

- Generation of solutions by ants according to private and pheromone information
- Update of the pheromone information

The implementation of these two steps is described below. A more specific description of the ACO algorithm is given in Doerner et al. (2001b).

A.1 Generation of solutions

The solution generation for the VRP can be described as follows. Given any partial solution, an ant (1) determines for each unassigned customer the best position to insert this customer into the partial solution. After that, the ant (2) probabilistically chooses one customer out of the set of unassigned customers and assigns this customer to the position determined in the first step. This procedure is repeated until the set of unassigned customers is empty. A decision step is depicted in Figure 1. The left graph shows the different options - shown as dashed lines - for the ant, the right graph depicts the situation after decision making. In this case the ant has chosen to go from customer 1 to customer 2.

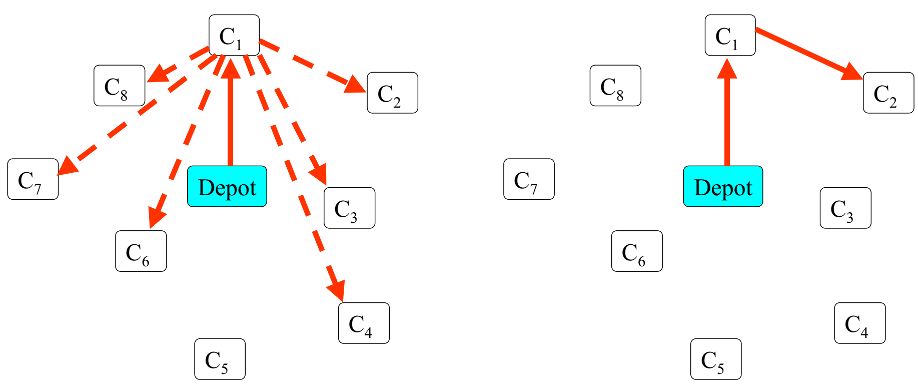


Fig. 1. An ants' decision

The decision rule is given by equation 1, where the matrices η and τ correspond to the local (heuristic) information and the pheromone information, respectively.

$$\mathcal{P}_{ij} = \begin{cases} q_0 + (1-q_0) \cdot \frac{[\tau_{ij}][\eta_{ij}]}{\sum_h [\tau_{ih}][\eta_{ih}]} & \text{if } \{[\tau_{ij}][\eta_{ij}]\} = max_h\{[\tau_{ih}][\eta_{ih}]\} \\ (1-q_0) \cdot \frac{[\tau_{ij}][\eta_{ij}]}{\sum_h [\tau_{ih}][\eta_{ih}]} & \text{otherwise.} \end{cases} \tag{1}$$

This decision rule is a combination of two mechanisms. The first one, exploitation, is achieved through a deterministic part in the rule, denoted by q_0. This factor constitutes the probability for choosing the deterministically best option as given in the conditional statement of the equation. By choosing the best option, the solution space near the global best solution is exploited. The second mechanism is a probabilistic choice according to the relative qualities of the available options.

A.2 Pheromone update

The pheromone update can be divided into two main categories.

First of all, pheromone evaporates due to two effects. One is a temporal effect, called global evaporation, which lowers the pheromone level on all paths. The other one is an effect that is intended to foster exploration. After an ant has traversed a given path, the pheromone level on this path is decreased. Thus, the other ants are less likely to follow that path. This effect is called local evaporation.

Second, some paths are reinforced, i.e. the pheromone level is increased. In particular, in our approach we follow a standard implementation, where after each iteration the paths, which constitute the global best solution, are reinforced.

This global pheromone update can be written as

$$\tau_{ij} = \begin{cases} (1-\rho)\tau_{ij} + \rho \cdot \frac{1}{L^*} & \text{if } (i,j) \in T^* \\ (1-\rho)\tau_{ij} & \text{otherwise.} \end{cases} \tag{2}$$

Here ρ denotes the percentage of the pheromone level that evaporates, L^* denotes the length of the global best solution and T^* is the representation of the global best solution. Thus, we see that the pheromone evaporates on all trails with a constant percentage, while the reinforcement is based on the quality of the solution found.

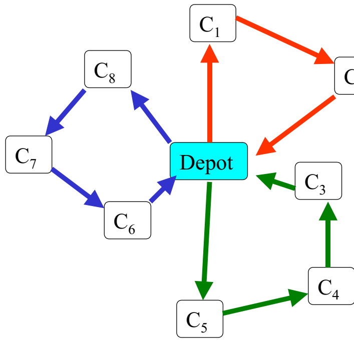
B. Genetic Algorithm

GAs represent a stochastic method of optimization based on the mechanisms of natural selection acting in nature(c.f. e.g. Gen and Cheng (1997), Mitchell (1996)). The notions of chromosome, gene and population constitute the base of GAs; and classical optimization theory terms of controlled variables vector, controlled variable and decision set can be brought into correspondence with them. The basic operations of a GA are crossover, mutation and selection. Crossover represents an operation on two parent-chromosomes yielding two offspring-chromosomes each of which inherits some of the genes from the parent-chromosomes. Mutation is a random gene modification. Selection represents itself as some procedure of population formation from the most adapted chromosomes. We propose to use a coding procedure and realizations of the above mentioned genetic operations for the VRP as follows.

B.1 GA coding

Any variant of decision is presented as a two-line string. The number of genes corresponds to the number of the customers. The first line of the chromosome represents a permutation, where customers are listed in the order in which they are visited and served. The second line indicates the clustering of customers. It contains only binary variables. A 0 indicates that this customer - the associated one in the first line of the string - is served first on a new tour, the 1's indicate that the associated customer is served on the same tour as the one to his left in the permutation.

For example, Figure 2 shows an exemplary solution and the corresponding chromosome.



Chromosome

1	2	8	7	6	5	4	3
0	1	0	1	1	0	1	1

Fig. 2. A feasible solution and its chromosome

Our suggested genetic coding allows us to use the same length chromosomes for solutions with different numbers of vehicles. The number of vehicles is just denoted by the number of zeros in the second line of the chromosome.

B.2 GA crossover

We propose to use a traditional one point crossover with a repair function, that exchanges duplicates of customers with missing customers. An example of our crossover is shown in Figure 3. The choice of parents for crossover is done using a canonical roulette wheel strategy (c.f. e.g. Gen and Cheng (1997)).

6	2	8	7	1	5	4	3	PARENT 1
0	1	0	1	1	0	1	1	
2	4	3	7	8	5	1	6	PARENT 2
0	1	1	0	1	1	0	1	
6	2	8	7	1	5	1	6	OFFSPRING 1 infeasible
0	1	0	1	1	1	0	1	
6	2	8	7	1	5	4	3	OFFSPRING 1 feasible
0	1	0	1	1	0	0	1	
2	4	3	7	8	5	4	3	OFFSPRING 2 infeasible
0	1	1	0	1	0	1	1	
2	4	3	7	8	5	6	1	OFFSPRING 2 feasible
0	1	1	0	1	0	1	1	

Fig. 3. An example of the crossover with the repair function

Let us consider offspring 1. Here, we repair the first line of the chromosome, by deleting duplicate customers and adding missing ones. In this case the first line of offspring 1 completely resembles parent 1. However, this need not always be the case, as can be seen by comparing offspring 2 with parent 2. Apart from that, having said that the first line of offspring 1 resembles parent 1, even in this case does not mean that the two chromosomes are the same. In fact, the difference occurs in the clustering, which is encoded in the second line.

As this clustering may lead to infeasible solutions too, we again need a repair mechanism. The dark grey shaded cells in offspring 1 show that repair. In fact an infeasible tour is repaired by flipping a bit from 1 to 0 in the second line. This means, that the infeasible tour is split up into two feasible tours. Thus, this repair always increases the number of tours.

B.3 GA mutation

There are two realizations of mutation operations suggested in the paper. The first one called permutation mutation exchanges customers between some tours without changing the number of vehicles. The second one called inversion mutation changes the number of vehicles. Individuals are randomly selected for mutation.

- Permutation mutation

Permutation mutation is realized as the following two-step procedure:
1. Choose two customers at random.
2. Exchange positions of chosen customers in the first line of the chromosome.

- Inversion mutation

Inversion mutation is realized as the following two-step procedure:
1. Choose a customer at random.
2. Inverse value of the gene in the second line at the chosen position.

Examples of these mutations are shown in Figure 4.

Permutation

2	4	3	5	1	8	7	6	Before mutation
0	1	1	0	1	1	0	1	
2	8	3	5	1	4	7	6	After mutation
0	1	1	0	1	1	0	1	

Inversion

2	4	3	5	1	8	7	6	Before mutation
0	1	1	0	1	1	0	1	
2	4	3	5	1	8	7	6	After mutation
0	1	0	0	1	1	0	1	

Fig. 4. Two types of mutations

B.4 GA selection

We use a deterministic strategy for selection. The selection procedure we use exhibits the strongest selection pressure. In each generation the best individuals survive, where parents and offspring compete for survival. We use a steady-state GA with constant population size.

C. The hybrid ACO-GA framework

We use two different populations, one of which is a colony of ants, while the other one is the population of a GA. Both populations solve the VRP iteratively, and after a pre-specified number of iterations, they will communicate their solutions. More specifically, the population with the superior solution reports this solution to the other population. More specifically, in our simulations presented below, we communicate solutions only if the global best solution is improved. This framework is shown in Figure 5.

We will test this new hybrid approach against independent approaches of ACO and GA. Thus, in section III we will present results for three approaches:

- ACO - this approach utilizes only a single population of ants
- GA - this approach utilizes only a single population of the Genetic Algorithm
- ACO+GA - this is our new hybrid approach

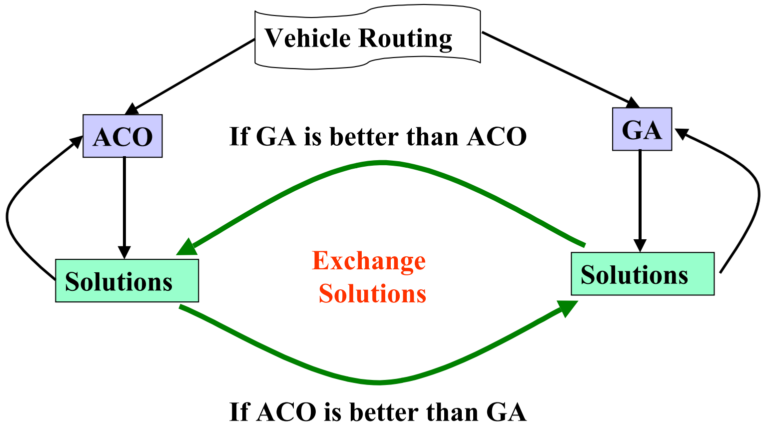


Fig. 5. Framework of the proposed solution method

III. Numerical analyis

Using a benchmark problem instance for the VRP we compared our new hybrid approach with independent ACO and GA approaches. The instance we used reflects a problem with 50 customers randomly distributed on the plane. These customers have to be served from a central depot. The capacity of the identical vehicles located at this central depot is 160 units. Together with the total demand of all customers, this leads to a minimal reqiurement of five vehicles. There is no constraint on the maximum length of a tour.

In Table I we summarized the results for ten runs of the three approaches[1]. The last four rows contain some statistics for the methods. The results show, that the GA working independently performs very poor. While this was somewhat surprising at first, we soon found that the poor performance was caused by the single point crossover we used. A more detailed comment on this problem together with some ideas for improvements is given in the next section.

Run	ACO	GA	ACO+GA
1	554,5311	653,5272	541,654
2	538,8215	848,9594	534,5318
3	531,4723	713,0126	530,1783
4	539,4921	618,0711	531,4723
5	528,9821	713,235	528,9821
6	544,4	698,8055	545,1514
7	561,7087	790,0409	557,1417
8	554,1111	751,2052	546,9176
9	549,9171	704,5667	549,6272
10	526,711	767,8561	530,1783
Avg.	543,0147	725,928	539,5835
Min	526,711	618,0711	528,9821
Max	561,7087	848,9594	557,1417
Std.dev.	17,51244	115,5272	14,22231

TABLE I

Comparison of the results for the three methods.

If we compare the results of the ACO with those of our hybrid approach, Table I shows no significant

[1]Note, that the GA consists of a population of $GApop = 40$ individuals, the ACO consists of $ACOpop = 10$ ants per iteration, while our hybrid approach utilizes $ACOpop = 10$ ants and $GApop = 30$ individuals in the GA. These differences where chosen to keep the computation times comparable. Furthermore we perform $2 \cdot GApop$ crossover operations per iteration and mutation occurs with a probability of 1%. In our ACO module, we use $q_0 = 0.9$ and $\rho = 0.1$.

difference between these two approaches. Both averages and standard deviations confirm this result.

Given these results we wanted to see whether the GA has any influence on the solutions found by the hybrid approach. Figure 6 shows the change of the solution quality over time for a typical run. In particular, the figure shows the global best solution found by the ACO and by the GA, respectively.

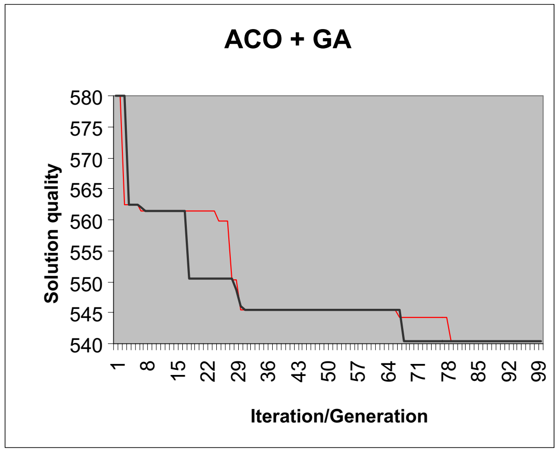


Fig. 6. A typical run of the hybrid method.

From Figure 6 we see that the GA sometimes indeed improves the solutions found by the ACO. However, this happens not very frequently. Most improvements are found by the ants. Moreover, while the ACO transfers a new global best solution 'as such' to the GA, the latter 'always' knows this global best solution. Given this fact, one would expect better performance of the GA, i.e. more frequent improvements. On the other hand, the GA transfers a new global best solution to the ants, by updating the pheromone matrix. As this matrix also contains information about the previous best solution, the ACO needs some time to 'understand' the new solution - the time it takes for the previous solution to evaporate sufficiently. In Figure 6 this is reflected by the extended periods, where the best solution found by the ACO lies above the global best solution.

IV. Conclusions and future research

In this paper we propose a new hybrid approach for solving a particular combinatorial optimization problem, namely the VRP. The approach is based on the combination of an ACO algorithm with a GA. Our preliminary results show, that the hybrid approach yields no additional benefit with respect to solution quality. However, we strongly believe that this is due to the rather poor performance of the GA. In fact, the single point crossover and the necessary repair operation for the chromosomes lead to a mutation-like crossover. The second part of the chromosome, coming from the second parent, generally looses any resemblance with the second parent.

While the hybrid approach does not outperform the independent ACO algorithm, the results show, that given a good solution, the GA is able - through mutation - to improve such a solution and thus support the ants to find good solutions more quickly. Thus, we strongly believe, that by improving the GA we will be able to show the benefit of our hybrid approach.

Our main idea for such an improvement, is to try and extract 'building blocks' of good solutions and generate offspring based on such building blocks. In the VRP such an approach could aim to recombine good individual tours found by the parents.

More generally, we will also aim to consider different ways of information exchange between the GA and the ACO.

Acknowledgments

This work was initiated during the Complex Systems Summer School 2001 in Budapest, jointly organized by the Santa Fe Institute and the Central European University. We would like to thank both institutions for financial and intellectual support. In particular, we thank Professor Melanie Mitchell for organizing the working group on 'Adaptive Computation'.

References

[1] Doerner, K. F., Hartl, R.F., and Reimann, M.: Cooperative Ant Colonies for Optimizing Resource Allocation in Transportation. In: Boers et al. (Eds.): Applications of Evolutionary Computing. Springer LNCS 2037, Berlin/Heidelberg (2001)

[2] Doerner, K. F., Hartl, R.F., and Reimann, M.: Are CompetANTS competent for problem solving - the case of a transportation problem. POM Working Paper 01/2001

[3] Gambardella, L. M., Taillard, E. and Agazzi, G.: MACS-VRPTW: A Multiple Ant Colony System for Vehicle Routing Problems with Time Windows. In: Corne, D., Dorigo, M. and Glover, F. (Eds.): New Ideas in Optimization. McGraw-Hill, London (1999)

[4] Garey, M. R. and Johnson, D. S.: Computers and Intractability: A Guide to the Theory of NP Completeness. W. H. Freeman & Co., New York (1979)

[5] Gen, M. and Cheng R.: Genetic algorithms and engineering design. John Wiley and Sons, New York (1997)

[6] Hansen, P. and Mladenovic, N.: An introduction to variable neighborhood search. In: Voss, S. et al. (Eds.): Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization. Kluwer Academic Publishers, Boston (1999)

[7] Mitchell, M.: An Introduction to Genetic Algorithms. MIT Press, Cambridge (1996)

[8] Rochat, Y. and Taillard, E. D.: Probabilistic Diversification and Intensification in Local Search for Vehicle Routing. Journal of Heuristics **1** (1995) 147–167