Applying the Ant System to the Vehicle Routing Problem

BERND BULLNHEIMER, RICHARD F. HARTL AND CHRISTINE STRAUSS

Department of Management Science, University of Vienna
Bruenner Str. 72, A-1210 Vienna, Austria
Email(s): {bullnhei, hartl, strauss}@pom.bwl.univie.ac.at

Abstract

In this paper we use a recently proposed metaheuristic, the Ant System, to solve the Vehicle Routing Problem (VRP) in its basic form, i.e. with capacity and distance restrictions, one central depot and identical vehicles. A "hybrid" Ant System algorithm is first presented and then improved using problem specific information (savings, capacity utilization). Experiments on various aspects of the algorithm and computational results for fourteen benchmark problems are reported and compared to those of other metaheuristic approaches such as Tabu Search, Simulated Annealing and Neural Networks.

1 Introduction

The Ant System, introduced by Colorni, Dorigo and Maniezzo [6], [10], [12] is a new distributed metaheuristic for hard combinatorial optimization problems and was first used on the well known Traveling Salesman Problem (TSP). It has further been applied to the Job Shop Scheduling Problem in [7], to the Graph Colouring Problem in [8] and to the Quadratic Assignment Problem in [18].

Observations on real ants searching for food were the inspiration to imitate the behaviour of ant colonies for solving combinatorial optimization problems. Real ants are able to communicate information concerning food sources via an aromatic essence, called pheromone. They mark the path they walk on by laying down pheromone in a quantity that depends on the length of the path and the quality of the discovered food source. Other ants can observe the pheromone trail and are attracted to follow it. Thus, the path will be marked again and will therefore attract more ants. The pheromone trail on paths leading to rich food sources close to the nest will be more frequented and will therefore grow faster.

The described behaviour of real ant colonies can be used to solve combinatorial optimization problems by simulation: artificial ants searching the solution space simulate real ants searching their environment, the objective values correspond to the quality of the food sources and an adaptive memory corresponds to the pheromone trails. In addition, the artificial ants are equiped with a local heuristic function to guide their search through the set of feasible solutions.

In this paper we present the application of the ant system to the Vehicle Routing Problem (VRP) with one central depot and identical vehicles. The remainder of the paper is organized as follows: In §2 we present the VRP and the ant system algorithm to tackle it. A "hybrid" ant system algorithm, using the 2-opt heuristic and problem specific information, is developed in §3 and §4, respectively. Experiments on various aspects of the algorithm and computational results for fourteen benchmark problems are presented in §5. We conclude with a discussion of the results in §6.

2 Ant System for VRPs

The Vehicle Routing Problem (VRP) can be represented by a complete weighted directed graph $G = (V, A, d)$ where $V = \{v_0, v_1, v_2, \ldots, v_n\}$ is a set of vertices and $A = \{(v_i, v_j) : i \neq j\}$ is a set of arcs. The vertex v_0 denotes the *depot*, the other vertices of V represent *cities* or *customers*, and the nonnegative weights d_{ij}, which are associated with each arc (v_i, v_j), represent the distance (or the travel time or the travel cost) between v_i and v_j. For each customer v_i a nonnegative demand q_i and a nonnegative service time δ_i is given ($q_0 = 0, \delta_0 = 0$). The aim is to find minimum cost vehicle routes where

- every customer is visited exactly once by exactly one vehicle
- all vehicle routes begin and end at the depot
- for every vehicle route the total demand does not exceed the vehicle capacity Q
- for every vehicle route the total route length (incl. service times) does not exceed a given bound L.

The VRP is a very complicated combinatorial optimization problem that has been worked on since the late fifties, because of its central meaning in distribution management. Problem specific methods (e.g. [5], [15]) as well as metaheuristics like tabu search (e.g. [13]), simulated annealing (e.g. [19]), genetic algorithms (e.g. [17]) and neural networks (e.g. [14]) have been proposed to solve it.

The VRP and the TSP are closely related. As soon as the customers of the VRP are assigned to vehicles, the VRP is reduced to several TSPs. For that reason our approach is highly influenced by the TSP ant system algorithm by Dorigo et al. [12].

To solve the VRP, the artificial ants construct vehicle routes by successively choosing cities to visit, until each city has been visited. Whenever the choice of another city would lead to an infeasible solution for reasons of vehicle capacity or total route length, the depot is chosen and a new tour is started. For the selection of a (not yet visited) city, two aspects are taken into account: how good *was* the choice of that city, an information that is stored in the pheromone trails τ_{ij} associated with each arc (v_i, v_j), and how promising *is* the choice of that city. This latter measure of desirability, called visibility and denoted by η_{ij}, is the local heuristic function mentioned above. In the case of the VRP (or the TSP) it is defined as the reciprocal of the distance, i.e. $\eta_{ij} = 1/d_{ij}$.

With $\Omega = \{v_j \in V : v_j \text{ is feasible to be visited}\} \cup \{v_0\}$, city v_j is selected to be vistited after city v_i according to a *random-proportional rule* [11] that can be stated as follows:

$$p_{ij} = \begin{cases} \frac{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta}{\sum_{h\in\Omega} [\tau_{ih}]^\alpha [\eta_{ih}]^\beta} & \text{if } v_j \in \Omega \\ 0 & \text{otherwise} \end{cases} \tag{1.1}$$

This probability distribution is biased by the parameters α and β that determine the relative influence of the trails and the visibility, respectively.

After an artificial ant k has constructed a feasible solution, the pheromone trails are laid depending on the objective value L_k. For each arc (v_i, v_j) that was used by ant k, the pheromone trail is increased by $\Delta\tau_{ij}^k = 1/L_k$. In addition to that, all arcs belonging to the so far best solution (objective value L^*) are emphasized as if σ ants, so-called *elitist ants* had used them. One elitist ant increases the trail intensity by an amount $\Delta\tau_{ij}^*$ that is equal to $1/L^*$ if arc (v_i, v_j) belongs to the so far best solution, and zero otherwise. Furthermore, part of the existing pheromone trails evaporates (ρ is the trail persistence)[1]. Thus, the trail intensities are updated according to the following Formula (1.2), where m is the number of artificial ants:

$$\tau_{ij}^{new} = \rho\tau_{ij}^{old} + \sum_{k=1}^{m} \Delta\tau_{ij}^k + \sigma\Delta\tau_{ij}^* \tag{1.2}$$

Concerning the initial placement of the artificial ants it was found that the number of ants should be equal to the number of cities in the TSP, and that each ant should start its tour from another city[2]. The implication for the VRP is that as many ants are used as there are customers in the VRP (i.e. $m = n$), and that one ant is placed at each customer at the beginning of an iteration. After initializing the basic ant system algorithm, the two steps *construction of vehicle routes* and *trail update*, are repeated for a given number of iterations.

3 "Hybrid" Ant System Algorithm

Hybridization in general means combining ideas of two different methods in one approach. Such proceeding is common practice for hard combinatorial optimization problems and has been successfully applied to other problems like e.g. timetabling [3] or production scheduling [20].

The *2-opt-heuristic* for the TSP [9] is an exchange procedure that generates a so-called *2-optimal* tour. A tour is called 2-optimal if there is no possibility to shorten the tour by exchanging two arcs. In vehicle routing 2-opt is used in the *Sweep-Algorithm* [15], where customers are first clustered and then 2-optimal vehicle routes for each cluster are generated. The same can be done with solutions constructed by artificial ants: at the end of an ant system iteration, each vehicle route generated is checked for 2-optimality and is improved if possible. Only then the total objective value is calculated and the trails are updated. The "hybrid" ant system[3] for the VRP can be described by the schematic algorithm given in Figure 3.1.

The resulting quantitative improvements achieved by the "hybrid" ant system are shown in §5. In the next section the "hybrid" ant system is further improved by including problem specific information in step II (a) of the algorithm, the construction of vehicle routes.

[1] Elitist ants improved the results obtained for the TSP and were therefore also used for the VRP. Trail evaporation is used to avoid early convergence. For a more detailed description the reader is referred to [12].

[2] These are results of experiments by Dorigo et al. [12] as well as our own experiments.

[3] It is questionable whether the addition of the 2-opt approach deserves the name hybrid method or whether it is only a post-optimization. We use the term *hybrid* following Goldberg's schematic of a "hybrid using a batch scheme" [16], p.203.

I	Initialize	
II	For I^{max} iterations do:	
	(a)	For each ant $k = 1, \ldots, m$ generate a new solution using Formula (1.1)
	(b)	Improve all vehicle routes using the 2-opt-heuristic
	(c)	Update the pheromone trails using Formula (1.2)

Figure 3.1 : "Hybrid" Ant System Algorithm

4 Problem Specific Improvements

The close relation between the VRP and the TSP, and thus the corresponding ant system approaches, has been mentioned above. One major difference, namely the existence of one distinct city in the VRP, the depot, has been taken into account. But the VRP has some further characteristics, that can be included in an ant system algorithm for the purpose of quality improvement of the solutions.

In the VRP not only the relative location of two cities is important[4] but also their relative location to the depot v_0 is essential for the tour length. The so-called *savings*[5] measure the favourability of combining two cities v_i and v_j to a tour and can be quantified by: $\mu_{ij} = d_{i0} + d_{0j} - d_{ij}$. High savings μ_{ij} indicate that visiting customer v_j after v_i is a good choice. This can be used to improve the quality of the ant system algorithm if high savings lead to a high probability of selection, i.e. if

$$p_{ij} \sim \mu_{ij}^{\gamma}$$

where the parameter γ regulates the relative influence of the savings.

Furthermore, for a capacity restricted problem as the VRP, it seems reasonable to assure a high degree of capacity utilization of the vehicles. Let Q_i the total capacity used including the capacity requirement of customer v_i, then high[6] values $\kappa_{ij} = (Q_i + q_j)/Q$ indicate high capacity utilization through the visit of customer v_j after visiting v_i. This again can be used for the ant system by giving those customers a high probability of being selected:

$$p_{ij} \sim \kappa_{ij}^{\lambda}$$

The parameter λ determines the relative influence of κ_{ij}. The probability distribution for selecting customer v_j to be visited next after customer v_i can thus be extended to:

$$p_{ij} = \begin{cases} \frac{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta [\mu_{ij}]^\gamma [\kappa_{ij}]^\lambda}{\sum_{h \in \Omega} [\tau_{ih}]^\alpha [\eta_{ih}]^\beta [\mu_{ih}]^\gamma [\kappa_{ih}]^\lambda} & \text{if } j \in \Omega \\ 0 & \text{otherwise} \end{cases} \qquad (1.1')$$

[4] This information is included in the visibility.
[5] See the *Savings-Algorithm* in [5]. That approach starts with depot-customer-depot tours. Then, according to decreasing savings, tours are combined as long as no restrictions are violated.
[6] $\kappa_{ij} \leq 1$ for a feasible solution.

5 Computational Results

The ant system for VRPs was tested on fourteen benchmark problems described in [4]. These problems contain between 50 and 199 customers in addition to the depot. The customers in problems C1-C10 are randomly distributed in the plane, while they are clustered in problems C11-C14. Problems C1-C5 and C6-C10 are identical, except that for the latter the total route length is bounded, whereas for the former it is not. The same is true for the clustered problems: problems C13-C14 are the counterparts of problems C11-C12 with additional route length constraint. For the problems with bounded route length all customers require the same service time $\delta = \delta_1 = \ldots = \delta_n$.

Before the results for these test problems are presented at the end of this section, we illustrate some of our experiments[7] as well as the stepwise improvement of the ant system at hand of problem C1, which contains 50 randomly distributed customers. As a summary of the results we present the deviation from the optimal solution[8] for the best and the average solution out of 30 runs in Table 5.1. The basic ant system

method	Ø	dev.	best	dev.
NN	646.22	23.18%	599.66	14.31%
AS	617.47	17.70%	590.74	12.61%
HAS	592.32	12.91%	564.44	7.59%
HAS-sav	554.36	5.67%	542.61	3.43%
HAS-cap	563.52	7.42%	542.85	3.48%
HAS-1	546.11	4.10%	532.88	1.58%
HAS-5	540.42	3.01%	524.61	0.00%

Table 5.1 : Comparison of Results

algorithm (denoted by AS in Table 5.1) solved problem C1 just satisfactory in the first experiment[9]. The best solution the ants found by selecting the customers according to the probability distribution given in Formula (1.1) was 12%, the average over 30 runs was 17% above the optimum. To see whether the pheromone trails contribute at all to the results we tested $\alpha = 0$, a setting that could be described as a *stochastic nearest neighbour heuristic* (denoted by NN). The results showed clearly that using the trail information does contribute to the quality of the solution: without it the average objective value was 23% above the optimum. The "hybrid" ant system (HAS) on the other hand, generated much better solutions (dev. 7%) than the basic ant system.

Through the problem specific features described in §4, i.e. through the use of Formula (1.1′) for the selection probabilities, a further reduction of route lengths was achieved. In three tests we studied the sole influence of respectively, savings

[7] For each experiment we simulated 30 independent ant system runs of 50 iterations each. As we used one ant per customer, the number of solutions generated per iteration was equal to the number of customers, thus a total of 2500 solutions was generated per run.

[8] Strictly speaking these solutions are not always the optimal but the "best published" solutions as only for some of them optimality has been proven. In the following there is no distinction made regarding this aspect.

[9] The parameter setting $\alpha = 1$, $\beta = 5$ and $\rho = 0.75$ lead to good results for the TSP (cf. Footnote 2) as well as the VRP and was chosen, if not indicated otherwise.

($\gamma = 5, \lambda = 0$, denoted by HAS-sav) and capacity utilization ($\gamma = 0, \lambda = 5$, HAS-cap), as well as their combined influence ($\gamma = \lambda = 5$, HAS-1)[10]. Both features improved the performance of the ant system algorithm, with the savings yielding better results (avg. dev. 5.6% as compared to 7.4%), and worked best when applied simultaneously (avg. dev. 4%). As a consequence of the reduced influence of the pheromone trails compared to visibility, savings and capacity utilization, the adaptive effect almost vanished. Therefore all terms were weighted equally and the parameter setting $\alpha = \beta = \gamma = \lambda = 5$ was chosen, which lead to the best results where the ant system (HAS-5) found the optimal solution (total length 524.61).

In order to analyze the ant specific contribution to the quality of the results, we further compared the "hybrid" ant system (HAS-5) with a stochastic local search procedure[11]. The latter uses visibility, savings and capacity utilization for tour construction ($\alpha = 0, \beta = \gamma = \lambda = 5$, i.e. no pheromone trails are used), and the 2-opt heuristic for tour improvement.

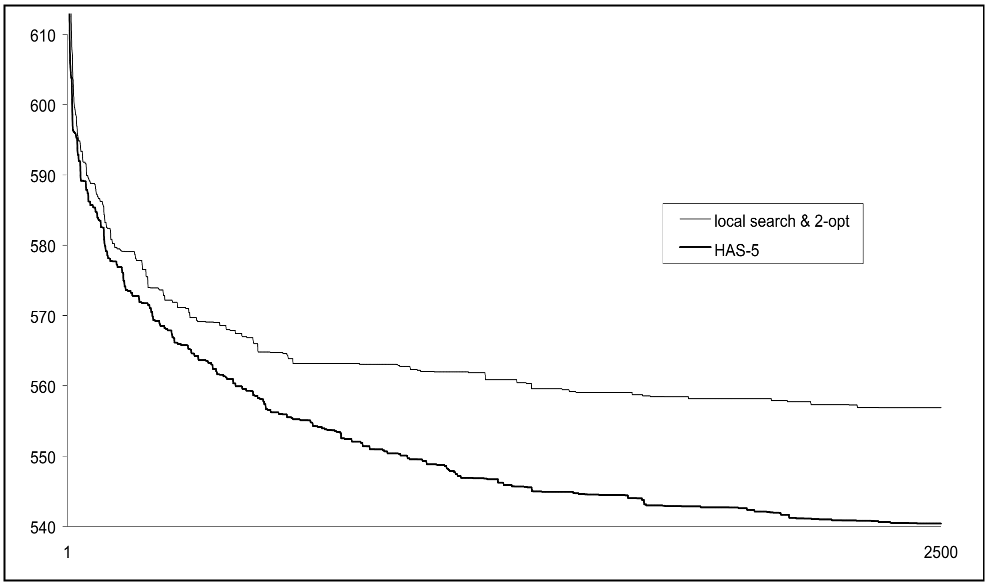

Figure 5.1: Ant System vs. Local Search

Figure 5.1 depicts the continuous reduction of objective values (50 iterations $\widehat{=}$ 2500 solutions, averaged over 30 runs) for both methods. The graph shows clearly that the *local search & 2-opt procedure* is outperformed by the "hybrid" ant system. In the early phase of the search the two methods look almost identical. The trail intensities are still close to their initial value τ_0 and have therefore hardly any effect on the selection probabilities. Thus, the artificial ants select the customers in this stage primarily according to visibility, savings and capacity utilization, which is also done in the

[10] The other two parameters were kept at $\alpha = 1$ and $\beta = 5$.

[11] Recall the similar comparison between the basic ant system (AS) and the stochastic nearest neighbour heuristic (NN).

local search procedure. Later, when trail intensities for some arcs increase because of frequent use, and decrease for others because of evaporation, the ants use this accumulated information. Thus, the solution space is reduced and better solutions are generated, whereas the local search is still based on initial data only.

σ	Ø	dev.	best	dev.
0	559.74	6.70%	552.04	5.23%
10	550.12	4.86%	528.20	0.68%
30	544.17	3.73%	525.13	0.10%
50	540.42	3.01%	524.61	0.00%
70	545.40	3.96%	530.26	1.08%
90	548.94	4.64%	531.84	1.38%

Table 5.2: Influence of Elitist Ants

In further tests we studied the influence of the elitist ants. In [12] the ant system performance for a TSP with 30 cities first increased with the number of elitist ants (up to an optimal range around 8) and then decreased again. For the VRP we found a similar phenomenon: introducing elitist ants and increasing their number brought better results, but only up to a range around 50, i.e. the number of "regular" ants / customers. The use of more elitist ants lead to poorer performance, caused by massive exploration of suboptimal tours early in the search. The results for various numbers of elitist ants are illustrated in Table 5.2.

initial placement	Ø	dev.	best	dev.
depot	550.84	5.00%	527.98	0.64%
customer	540.42	3.01%	524.61	0.00%
random	545.76	4.03%	531.90	1.39%

Table 5.3: Influence of Initial Placement

Furthermore we looked at the initial placement of the artificial ants. As the VRP has one distinct city, namely the depot, starting the search from there is another possibility because the depot is per definition included in every vehicle route. Alternatively, choosing the starting points for the artificial ants randomly is also possible. The comparison of these options, which is illustrated in Table 5.3, confirms our assumption that placing one ant at each customer is best.

Finally the influence of the trail persistence was subject of further tests (cf. Table 5.4). The results underline our early findings that $\rho = 0.75$ is a good setting. Higher values prevent efficient exploration of the search space as the trail intensities on arcs belonging to suboptimal vehicle routes are kept too high for too long. For lower values the learning effect diminishes and even though the finding of very good solutions is possible, the average quality of the algorithm decreases.

The following Table 5.5 compares the computational results for the fourteen test

ρ	Ø	dev.	best	dev.
0.99	545.68	4.02%	531.66	1.34%
0.95	544.33	3.76%	525.13	0.10%
0.75	540.42	3.01%	524.61	0.00%
0.50	544.41	3.77%	524.63	0.00%
0.25	548.42	4.54%	524.93	0.06%

Table 5.4 : Influence of Trail Persistence

Random problems								
Prob.	n	Q	L	δ	optimal solution	Ant System	dev.	vehicles used
C1	50	160	∞	0	524.61[a]	524.61	0.00%	5
C2	75	140	∞	0	835.26[a]	870.58	4.23%	10
C3	100	200	∞	0	826.14[a]	879.43	6.45%	8
C4	150	200	∞	0	1028.42[a]	1147.41	11.57%	12
C5	199	200	∞	0	1291.45[b]	1473.40	14.09%	16
C6	50	160	200	10	555.43[a]	562.93	1.35%	6
C7	75	140	160	10	909.68[a]	948.16	4.23%	12
C8	100	200	230	10	865.94[a]	886.17	2.34%	9
C9	150	200	200	10	1162.55[a]	1202.01	3.39%	14
C10	199	200	200	10	1395.85[b]	1504.79	7.80%	19
Clustered problems								
Prob.	n	Q	L	δ	optimal solution	Ant System	dev.	vehicles used
C11	120	200	∞	0	1042.11[a]	1072.45	2.91%	9
C12	100	200	∞	0	819.56[a]	819.96	0.05%	10
C13	120	200	720	50	1541.14[a]	1590.52	3.20%	12
C14	100	200	1040	90	866.37[a]	869.86	0.40%	11

[a] Taillard [24]

[b] Rochat and Taillard [22]

Table 5.5 : Ant System Results

problems. For each problem the columns give the problem size n, the vehicle capacity Q, the maximal route length L, the service time δ and the objective value of the optimal solution. In the last three columns the best solutions obtained with the ant system, the deviation from the optimum and the number of vehicles used are shown. According to our findings we set $\rho = 0.75$ and used $m = n$ ants, initially placed at the customers $v_1, \ldots, v_n$. For all problems $I^{max} = 100$ iterations were simulated with $\sigma = n$ elitist ants. The random problems were solved using HAS-5 ($\alpha = \beta = \gamma = \lambda = 5$). For the problems C11-C14, where the customers are clustered, we found that the savings do not really contribute to an improvement. The reason is that cities belonging to different clusters, which are located behind each other, might be combined to a tour because of high savings (which result from being located in line with the depot). Thus, we used HAS-cap ($\alpha = 1, \beta = 5, \gamma = 0$ and $\lambda = 5$) for the clustered problems.

The computational results show that reasonably good solutions can be obtained by the ant system. Especially the results on the clustered problem instances C11-C14 seem to be better. There the deviation from the optimum ranged from 0.05% to 3.20%. Most random problems were solved within a 5% range, only for problems C4 and C5 the ant system showed higher deviations.

As run times are another criteria for the quality of an algorithm the proposed method is compared to other metaheuristic approaches for which run times were reported in Table 5.6. Tabu search (the sequential algorithm from [21][12]) outperforms

Random problems								
Prob.	Tabu Search [21]		Simulated Annealing [19]		Neural Networks [14]		Ant System	
C1	0.00%	0.9	0.65%	0.1	2.78%	0.9	0.00%	0.6
C2	0.27%	16.8	0.40%	59.4	—	—	4.23%	2.4
C3	0.17%	33.9	0.37%	102.9	8.14%	6.5	6.45%	11.3
C4	2.52%	27.2	2.88%	71.6	5.47%	13.2	11.57%	28.5
C5	3.64%	16.3	6.55%	22.9	8.51%	23.2	14.09%	82.2
C6	0.00%	3.2	0.00%	11.6	1.06%	4.3	1.35%	0.2
C7	0.00%	23.1	0.00%	5.2	—	—	4.23%	3.5
C8	0.27%	8.6	0.09%	6.1	3.28%	18.4	2.34%	7.3
C9	1.40%	15.6	0.14%	983.6	8.73%	27.2	3.39%	26.6
C10	1.79%	52.0	1.58%	40.3	13.22%	52.4	7.80%	57.3
Clustered problems								
Prob.	Tabu Search [21]		Simulated Annealing [19]		Neural Networks [14]		Ant System	
C11	0.14%	6.3	12.85%	4.4	5.79%	4.2	2.91%	16.2
C12	0.00%	1.2	0.79%	0.8	0.68%	1.7	0.05%	10.1
C13	0.59%	2.0	0.31%	76.2	4.37%	31.3	3.20%	4.3
C14	0.02%	9.4	2.73%	5.0	1.55%	8.5	0.40%	3.1
Ø	**0.77%**		**2.09%**		**5.30%**		**4.43%**	
	Sun Sparc 4		VAX 8600		VAX 8600		Pentium 100	

Table 5.6: Deviation and Run Times for several Metaheuristic Approaches

all other metaheuristics with an average deviation of 0.77%. The ant system (4.43%) performs not as good as Osman's simulated annealing approach [19] with 2.09% but better than Ghaziri's neural networks approach [14] where the average deviation was 5.30% with only 12 out of 14 problems tested. Run times (given in CPU minutes in

[12] A comparison on basis of run times on different machines is not perfectly meaningful. To ensure maximum comparability we did not include their parallel implementation.

Table 5.6) for all algorithms are more or less similar and vary with the problem size in a range from approximately one minute for the smallest to approximately one and a half hours for the largest problem.

6 Discussion and conclusion

The presented contribution shows the application and the improvement of an ant system algorithm to the VRP. The computational results confirm the positive experiences made with the ant system by applying it to the TSP [1], [11], [23]. Although some very good solutions for the VRP instances were obtained, the best known solutions for the fourteen test problems could not be improved. For practical purposes deviations up to 5% are more than acceptable as uncertainty about travel costs, demands, service times etc. makes perfect planning impossible. As the ant system can compete with other vehicle routing metaheuristics in terms of run times, the presented approach is an alternative to tackle VRPs.

Tabu Search performs much better, but nevertheless the results for the ant system also indicate that there still is much potential for improvement. The superiority of tabu search for VRPs can be explained by two facts: tabu search is an excellent method that has been studied and improved a lot since its introduction, and, much more VRP related research has been done on tabu search (cf. [13], [19], [21], [22], [24]) than on any other method. Therefore we are certain that future work on the ant system approach will help to further improve its quality for vehicle routing, even though our current version can not yet compete with the best tabu search algorithms.

Primarily, a more detailed analysis of parameter values is necessary. A metaheuristic could be used to guide the search through the parameter space. Also an automatic adjustment of the parameters like done e.g. in Evolution Strategies might be of use for the ant system. In addition to that, more elaborated local search procedures exchanging customers not only within but also among tours should be looked at. Another very interesting aspect is the use of candidate lists. In the current version of the ant system all feasible customers have the chance to be selected. For many of them the probability of being selected is very low because of large distances, low trail levels or both. Concentrating only on the more promising candidates should yield better results. Moreover the algorithm seems to be well suited for parallel implementation [2].

A more radical change of the existing algorithm would be to use the ants only to cluster the customers and subsequently, to apply a local search to find good tours among them. A similar idea using a genetic algorithm as a cluster builder has been proposed in [17].

Besides these methodological considerations, additional modifications of the algorithm to extensions of the VRP, e.g. multiple depots or problems with time windows are of interest.

Acknowledgements

The authors would like to thank Marco Dorigo, Vittorio Maniezzo and Gerhard Waescher as well as two anonymous referees for their comments that helped to improve the quality of this paper.

Bibliography

[1] B. Bullnheimer, R.F. Hartl, and C. Strauss. A new rank based version of the ant system: a computational study. *Working Paper No.1, SFB Adaptive Information Systems and Modelling in Economics and Management Science*, Vienna, 1997.

[2] B. Bullnheimer, G. Kotsis, and C. Strauss. Parallelization Strategies for the Ant System. Paper presented at *Conference on High Performance Software for Nonlinear Optimization: Status and Perspectives (HPSNO'97)*, Ischia (Italy), 4-6 June 1997.

[3] E.K. Burke, D.G. Elliman, and R.F. Weare. A Hybrid Genetic Algorithm for Highly Constrained Timetabling Problems. In *Proc. 6-th Int. Conf. Genetic Algorithms (ICGA'95)*, pages 605-610, Morgan Kaufmann, 1995.

[4] N. Christofides, A. Mingozzi, and P. Toth. The Vehicle Routing Problem. In N. Christofides, A. Mingozzi, P. Toth, and C. Sandi, editors, *Combinatorial Optimization*, pages 315-338, Wiley, 1979.

[5] G. Clarke, and J.W. Wright. Scheduling of Vehicles from a Central Depot to a Number of Delivery Points. *Oper. Res.* 12 (1964), pages 568-581.

[6] A. Colorni, M. Dorigo, and V. Maniezzo. Distributed Optimization by Ant Colonies. In F. Varela, and P. Bourgine, editors, *Proc. Europ. Conf. Artificial Life (ECAL'91)*, pages 134-142, Elsevier Publishing, 1991.

[7] A. Colorni, M. Dorigo, V. Maniezzo, and M. Trubian. Ant system for Job-Shop Scheduling. *JORBEL - Belgian Journal of Operations Research, Statistics and Computer Science* 34 (1994) 1, pages 39-53.

[8] D. Costa, and A. Hertz. Ants can colour graphs. *J. Oper. Res. Soc.* 48 (1997), pages 295-305.

[9] G.A. Croes. A Method for solving Traveling-Salesman Problems. *Oper. Res.* 6 (1958), pages 791-812.

[10] M. Dorigo. *Optimization, Learning and Natural Algorithms.* Doctoral Dissertation, Politecnico di Milano, Italy (in Italian), 1992.

[11] M. Dorigo, and L.M. Gambardella. Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem. *IEEE Trans. Evol. Comput.* 1 (1997) 1, pages 53-66.

[12] M. Dorigo, V. Maniezzo, and A. Colorni. Ant System: Optimization by a Colony of Cooperating Agents. *IEEE Trans. Sys., Man, Cybernetics* 26 (1996) 1, pages 29-41.

[13] M. Gendreau, A. Hertz, and G. Laporte. A Tabu Search Heuristic for the Vehicle Routing Problem. *Management Sci.* 40 (1994), pages 1276-1290.

[14] H. Ghaziri. Supervision in the Self-Organizing Feature Map: Application to the Vehicle Routing Problem. In I. Osman, and J. Kelly, editors, *Meta-Heuristics: Theory & Applications*, pages 651-660, Kluwer Academic Publishers, 1996.

[15] B.E. Gillett, and L.R. Miller. A Heuristic Algorithm for the Vehicle Dispatch Problem. *Oper. Res.* 22 (1974) pages 340-347.

[16] D. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison-Wesley, 1989.

[17] H. Kopfer, G. Pankratz, and E. Erkens. Entwicklung eines hybriden Genetischen Algorithmus zur Tourenplanug. *Oper. Res. Spekt.* 16 (1994), pages 21-31.

[18] V. Maniezzo, A. Colorni, and M. Dorigo. The Ant System applied to the Quadratic Assignment Problem. *Technical Report IRIDIA / 94-28*, Universite Libre de Bruxelles, Belgium, 1994.

[19] I. Osman. Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. *Ann. Oper. Res.* 41 (1993), pages 421-451.

[20] E. Pesch. *Learning in Automated Manufacturing.* Physica, 1994.

[21] C. Rego, and C. Roucairol. A Parallel Tabu Search Algorithm Using Ejection Chains for the Vehicle Routing Problem. In I. Osman, and J. Kelly, editors, *Meta-Heuristics: Theory & Applications*, pages 661-675, Kluwer Academic Publishers, 1996.

[22] Y. Rochat, and E. Taillard. Probabilistic Diversification and Intensification in Local Search for Vehicle Routing. *J. Heuristics* 1 (1995), pages 147-167.

[23] T. Stuetzle, and H. Hoos. The MAX-MIN Ant System and Local Search for the Traveling Salesman Problem. *Proc. ICEC'97 - 1997 IEEE 4-th Int. Conf. Evolutionary Computation*, IEEE Press, pages 308-313.

[24] E. Taillard. Parallel Iterative Search Methods for Vehicle Routing Problems. *Networks* 23 (1993), pages 661-673.