Central European Journal of Operations Research manuscript No.
(will be inserted by the editor)

Heuristic Solution of an Extended Double-Coverage Ambulance Location Problem for Austria

Karl F. Doerner[1]**, Walter J. Gutjahr**[2]**, Richard F. Hartl**[1]**, Michaela Karall**[1]**, Marc Reimann**[3]

[1] Institute for Management Science, University of Vienna, Bruenner Strasse 72, 1210 Vienna, Austria

[2] Department of Statistics and Decision Support Systems, University of Vienna, Universitaetsstrasse 5, 1010 Vienna, Austria

[3] Institute for Operations Research, ETH Zurich, Clausiusstrasse 47, 8092 Zurich, Switzerland

Received: 11/01/2005 / Revised version: 04/04/2005

Abstract In this paper, we present solution procedures to tackle an ambulance location problem in Austria. We consider the problem as a double-coverage ambulance location problem, and for specifying it in formal terms, we use an extension of a model developed by Gendreau, Laporte and Semet [11] by introducing a limit on the number of inhabitants served per ambulance. To solve the problem, we reimplemented the Tabu Search algorithm of [11] and developed an Ant Colony Optimization algorithm. Our computational tests are based on real location sites, the population census data and the real road network data of eight provinces of Austria. For the two smallest instances exact solutions can be found and used as a measure for the performance of the two metaheuristic algorithms. For these problem instances, both metaheuristics find the optimal solution easily. For the larger instances, Tabu Search and Ant Colony Optimization yield results of comparable quality. However, Tabu Search turns out as distinctively faster.

Keywords: *Ant Colony Optimization, Tabu Search, Facility Location, Maximum Coverage Location Problem.*

1 Introduction

In the last few years, increasing cost pressure on not-for-profit health care organizations has led to the need for more efficient and effective provision of services. One of the most important cost factors is associated with the management of the emergency fleet. Here both fixed and variable fleet costs as well as personnel costs have to be covered. These costs depend on the number and spatial distribution of ambulances used, as well as on the control of the fleet through deployment and repositioning decisions.

In this paper, we will deal with the tactical planning problem of locating a given fleet of ambulances in an area, such that the service level is maximised. Here, service level is measured as the coverage of demands by the available fleet. The concept of coverage has been defined in various ways in the academic literature (see, e.g., [4], [6], [11] and [13]). An overview of different ambulance location and relocation models can be found in [3]. See also the general survey on location models in [15].

Our model follows and extends the model proposed by Gendreau, Laporte and Semet in [11]. In their model, a demand is said to be covered if it can be reached by an ambulance within a given, user-defined time limit. Both, single coverage (by exactly one ambulance vehicle) and double coverage (by at least two different ambulance vehicles) are considered. Further, two different time limits are set and the objective is to maximise double coverage within the smaller time limit. The main constraints imposed are that all demand has to be covered within the larger time limit, and that a certain percentage of the total demand has to be satisfied even within the smaller time limit. In this model it is assumed that all nodes have equal demands. Our extension takes the density of the population in different demand nodes into account and aims at solutions for which, within the larger time limit, a certain ratio between the number of inhabitants and the number of available ambulances is not exceeded. The intuition of this extension is to obtain more equity in service provision over all potential demand nodes.

We specified our model based on problem information and real data from Austria published by the Austrian Red Cross and provided by WigeoGis and TeleAtlas. For solving the model, we implemented two meta-heuristics based on Tabu Search (TS) and Ant Colony Optimization (ACO), respectively. In fact, the TS used is a re-implementation (and adaptation to our extended problem formulation) of the algorithm originally proposed in [11]. Computational results for the two metaheuristics for problem instances corresponding to eight provinces of Austria are presented. For the two smallest instances, exact solutions are found and used as benchmark for the performance of the two metaheuristics.

The main contributions of the paper are the extension of the double-coverage model to include different demand densities at the nodes as well as the proposition of ACO as a viable alternative to TS for tackling the extended model.

The remainder of this paper is organised as follows. In the next section, we present our model in detail. In Section 3, the two meta-heuristics are described, and results of our computational case study are shown in Section 4. Finally, Section 5 concludes with a summary of the paper and an outlook on possible future research.

2 The Model

As mentioned above, our model was developed to solve the double coverage problem for a given set of locations and a given number of ambulances. The objective is to maximise the demand covered by two ambulances within a small radius $r > 0$, while all demands have to be covered by at least one ambulance within a larger radius $R > r$.

First of all, we describe the double coverage problem as formulated in [11]; after that, our modification will be described. The problem is defined on a graph $G = (V \cup W, E)$ where $V = \{v_1, ..., v_n\}$ and $W = \{v_{n+1}, ..., v_{n+m}\}$ are two vertex sets representing demand points and potential location sites, respectively, and $E = \{(v_i, v_j) : v_i \in V \text{and } v_j \in W\}$ is the considered edge set. With each edge (v_i, v_j), a travel time t_{ij} is associated. The demand at vertex $v_i \in V$ (number of inhabitants in v_i) is equal to λ_i. The total number of ambulances is given and equal to p. For $v_i \in V$ and $v_{n+j} \in W$, the coefficients

$$\gamma_{ij} = \begin{cases} 1 & t_{i,n+j} \leq r \\ 0 & \text{otherwise} \end{cases} \qquad \delta_{ij} = \begin{cases} 1 & t_{i,n+j} \leq R \\ 0 & \text{otherwise} \end{cases}$$

are defined, where γ_{ij} (δ_{ij}) indicates whether or not demand node i is covered by location $n + j$ within the radius r (R).

A further input parameter ω gives the proportion of the total demand that must be covered by an ambulance within the small radius r. In our experiments, we used the same value for ω as was done in [11], i.e. $\omega = 0.8$. Moreover, for each location v_{n+j}, the integer p_j indicates the maximum number of ambulances that can be located at v_{n+j}.

For the problem formulation, we use the following decision variables: 1) variable z_j is of integer type and denotes the number of ambulances located at $v_{n+j} \in W$ and 2) the variables x_i and y_i are binary and denote whether or not a demand node i is covered within the small radius r at least once and at least twice, respectively.

Based on these variable definitions, the problem is given by

$$\max\ f(z) = \sum_{i=1}^{n} \lambda_i y_i \tag{1}$$

subject to the constraints

$$\sum_{j=1}^{m} \delta_{ij} z_j \geq 1 \ (v_i \in V) \tag{2}$$

$$\sum_{i=1}^{n} \lambda_i x_i \geq \omega \sum_{i=1}^{n} \lambda_i \tag{3}$$

$$\sum_{j=1}^{m} \gamma_{ij} z_j \geq x_i + y_i \ (v_i \in V) \tag{4}$$

$$y_i \leq x_i \ (v_i \in V) \tag{5}$$

$$\sum_{j=1}^{m} z_j = p \tag{6}$$

$$z_j \leq p_j \ (v_{n+j} \in W) \tag{7}$$

$$x_i, y_i \in \{0, 1\} \ (v_i \in V) \tag{8}$$

$$z_j \text{ integer } (v_{n+j} \in W). \tag{9}$$

The objective represents the maximization of the total demand covered at least twice within r. Note that the variables x_i and y_i can be computed from the variables z_j, such that the objective can be written as a function of z. Constraints (2) and (3) express single and double coverage requirements. Constraints (2) ensure that all demand is covered within R distance units. Constraints (3) ensure that a proportion ω of all demand is covered within the small radius r. Constraints (4) link the coverage of the demand nodes with the assignment of ambulances. The left-hand side of constraints (4) counts the number of ambulances covering v_i within the small radius r, while the right-hand side represents the level of coverage of v_i: it is equal to 1 if v_i is covered exactly once within the small radius r, and equal to 2 if it is covered at least twice within the small radius r. Constraints (5) ensure that a vertex v_i cannot be covered at least twice when it is not covered at least once. Constraints (6) and (7) impose limits on the maximum number of ambulances located over all potential locations and on each single location, respectively. Finally, constraints (8) and (9) are the usual binary and integrality requirements for the decision variables.

Our modification of this model consists of two steps: First of all, we turn the hard constraints (2) and (3) into soft constraints, represented by additional weighted penalty terms in the objective function. Note that by choosing sufficiently large values for the weights, this first modification gets equivalent to problem (1) – (9). Thus, our modification is a generalization of the above shown problem. However, we rather tried to choose suitable weights from the viewpoint of practical application.

Then, as mentioned in the introduction, we use information about the density of demand to balance the assignment of ambulances with respect to the covered demand. In the standard models available, a solution may be an assignment of ambulances to locations where some ambulances have to cover a large demand within the radius R, while some others cover only a small demand. Such an assignment can be unrealistic, since the capacity of the ambulances assigned to a large population may be insufficient to perform the required emergency services. Therefore, we add a penalty term to our objective function to avoid such assignments: For each demand node v_i, the number of inhabitants w_i per ambulance assigned within radius R is computed. If this quotient exceeds a given limit w_0, a penalty proportional to $(w_i - w_0)$ is subtracted from the objective function. In this way, we obtain the following extended problem

$$\max F(z) = f(z) - M_1 f_1(z) - M_2 f_2(z) - M_3 f_3(z) \tag{10}$$

subject to (4) – (9), where $f(z)$ is given by (1),

$$f_1(z) = |\{v_i \in V : \sum_{j=1}^{m} \delta_{ij} z_j = 0\}|, \tag{11}$$

$$f_2(z) = \omega - \min\{\omega, \left(\sum_{i=1}^{n} \lambda_i x_i\right) / \left(\sum_{i=1}^{n} \lambda_i\right)\}, \tag{12}$$

and

$$f_3(z) = \sum_{i=1}^{n} (w_i - w_0)^+ \tag{13}$$

with

$$w_i = \frac{\lambda_i}{\sum_{j=1}^{m} \delta_{ij} z_j}.$$

The function $f_1(z)$ counts the number of demand points *not* covered within the larger radius R. The function $f_2(z)$ represents the negative deviation of the degree of coverage within the smaller radius r from the intended level ω. The function $f_3(z)$ reflects the penalties for the violation of our additional constraints concerning the number of inhabitants per assigned ambulance. Finally, the weights $M_1 > 0$, $M_2 > 0$ and $M_3 > 0$ determine the relative importance of violations of the three soft constraints.

3 Solution approaches

In order to solve the model presented in the last section, we have implemented two metaheuristics based on Tabu Search and Ant Colony Optimization. A metaheuristic solution approach has been chosen not only in view of the fact that for larger instances exact solutions of the given problem cannot be computed anymore within reasonable time, but also in order to provide more flexibility for a fine-tuning of the model (extension of objective function and constraints) during the next planned phase of real-life implementation.

The TS is a reimplementation of the original algorithm in [11], while our ACO implementation is inspired by the ACO variant proposed for activity crashing problems in [7].

3.1 Approach 1: Ant Colony Optimization

The Ant Colony Optimization (ACO) metaheuristic introduced by Colorni, Dorigo and Maniezzo [5] and [9] imitates the behaviour shown by real ants when searching for food. Ants exchange information about food sources via pheromone, a chemical substance which they deposit on the ground as they move along. Short paths from the nest to a food source obtain a larger amount of pheromone per time unit than longer paths. As a result, an ever increasing number of ants is attracted to follow such routes, which in turn reinforces the corresponding pheromone trails. Artificial ants not only imitate the learning behavior described above, but also use additional, problem specific heuristic information.

ACO has been successfully applied to various hard combinatorial optimization problems (c.f. e.g. [10]), and convergence of specific ACO algorithms to optimal solutions has been shown in [12]. With the exception of[8], where an ACO approach has been used for a certain combined location-routing problem, ACO algorithms have, to our best knowledge, not been applied to location problems so far. Thus, we developed an ACO-based heuristic for the problem at hand. To exploit available knowledge, we used ideas from an ACO approach developed in [7] for the activity crashing problem, which has a solution set structure similar to that of the location problem under consideration. Our ACO algorithm ACOLoc described below builds on the solution construction mechanism for the best-performing ACO variant presented in [7].

3.1.1 Realization of the ACO Algorithm for Location Problems Figure 1 shows the pseudocode for the ACO algorithm for Location Problems (ACOLoc).

This algorithm works as follows: In an initialization phase, a number *popsize* of conceptual units called "ants" are generated. The iterative part of the ACO algorithm can be divided into two phases, a construction phase and a pheromone update phase. In the construction phase of the algorithm, each ant constructs a solution z by applying a pseudo-random-proportional rule (see Subsection 3.1.4), using attractiveness information η_{is} and pheromone information τ_{is}. The notions "attractiveness" and "pheromone" will be explained in Subsection 3.1.2 and 3.1.3, respectively. Finally, in the pheromone update phase, a pheromone update takes place, which will be described in Subsection 3.1.5.

In the solution construction part of the algorithm, each ant sequentially visits all locations $v_i \in W$ (in a random order) and assigns ambulances to these locations. Once p ambulances have been assigned, the remaining locations are not assigned any ambulances. Besides that, also the opposite case can occur where, after visiting all locations, not all p ambulances have been assigned yet. In this case the remaining ambulances are distributed to the locations by a simple local search procedure.

3.1.2 Attractiveness Information In ACO, "attractiveness" or "visibility" denotes a heuristic measure of how good a certain construction step will probably be. The attractiveness information is stored in a matrix η. The rows of the matrix correspond to the different locations $v_{n+j} \in W$, and the columns correspond to the number of ambulances s for each location v_{n+j}. The value η_{js} represents the attractiveness information for assigning s vehicles to location v_{n+j}.

For our application, the attractiveness values are bounded by

$$\eta_{min} \leq \eta_{js} \leq 1, \tag{14}$$

where the minimum attractiveness value η_{min} is a small constant. After some experimentation we found a value of $\eta_{min} = 0.001$ as a reasonable value. The attractiveness values are computed as follows. First, we compute for each potential location v_{n+j} the total amount of demand located within the small radius r. Then, this total demand is divided by the maximum demand w_0 that can be served by an ambulance, thus leading to the minimum number of ambulances s_{n+j} needed to cover all the demands within r from location v_{n+j}. According to this value s_{n+j}, the attractiveness values are computed as

$$\eta_{js} = \begin{cases} 1 & \text{if } s = s_{n+j} \\ \max\left\{\eta_{min}, 1 - 0.1 \cdot |s - s_{n+j}|\right\} & \text{otherwise.} \end{cases} \tag{15}$$

3.1.3 Pheromone Information The pheromone information is used to store information that has been obtained from experience with already evaluated solutions from previous iterations. The pheromone information is stored in a matrix τ. The rows of this matrix correspond to the locations v_{n+j} and the columns correspond to the different possible numbers of vehicles $s = 0, ..., \max_j p_j$. The value τ_{js} represents the current pheromone information to locate s ambulances at the location v_{n+j}.

The initial pheromone values are computed in a similar way as the attractiveness values, i.e.

$$\tau_{js} = \begin{cases} 1 & \text{if } s = s_{n+j} \\ \max\left\{\tau_{min}, 1 - 0.1 \cdot |s - s_{n+j}|\right\} & \text{otherwise.} \end{cases} \tag{16}$$

The minimum pheromone value τ_{min} is a small constant ($\tau_{min} = 0.001$).

3.1.4 Decision Rule Given the attractiveness and the pheromone information, ACOLoc chooses the number of vehicles s for the current location v_{n+j} probabilistically according to the following pseudo-random-proportional rule:

$$s = \begin{cases} \arg\max\{\tau_{js}^{\alpha} \cdot \eta_{js}^{\beta} : s = 0, \ldots, p_j\} & \text{if } q \leq q_0 \\ \hat{s} & \text{otherwise,} \end{cases} \tag{17}$$

where q is a random number uniformly distributed in $[0, 1]$, and q_0 is a parameter ($0 \leq q_0 < 1$) representing the probability that the number of vehicles s with the highest product of pheromone and visibility for location v_{n+j} is selected deterministically. Given that the drawing of q results in a value such that $q > q_0$, the random variable $\hat{s}$, which corresponds to the number of vehicles assigned to location v_{n+j}, is selected according to the following probability distribution

$$\mathcal{P}_{\hat{s}}(j) = \frac{\tau_{j\hat{s}}^{\alpha} \cdot \eta_{j\hat{s}}^{\beta}}{\sum_{h=0}^{p_j} \tau_{jh}^{\alpha} \cdot \eta_{jh}^{\beta}} \qquad (\hat{s} = 0, \ldots, p_j). \tag{18}$$

The given probability distribution is biased by the parameters α and β, which determine the relative influence of the pheromone and the visibility, respectively.

3.1.5 Pheromone Update A local pheromone update is performed once an artificial ant has visited all the possible locations v_{n+j}. Then, pheromone values τ_{js} are decreased for the selected number of ambulances s for each location v_{n+j} applying the local pheromone update rule

$$\tau_{js} = (1 - \rho) \cdot \tau_{js} + \rho \cdot \tau_{min}\,, \tag{19}$$

where τ_m is a lower bound of pheromone values (a small constant value), and ρ is the evaporation rate ($0 \leq \rho \leq 1$). On account of local updating, ants prefer to assign a number of ambulances that has not yet been chosen. As a result, the diversity of the solutions is enhanced.

The global pheromone update takes place right after all ants of a population have proposed solutions for the location of the ambulances, feasibility and solution quality have been determined and the local search has been performed. The update rule for the global update is

$$\tau_{js} = (1 - \rho) \cdot \tau_{js} + \rho \cdot \Delta\tau_{js} \cdot 2\,. \tag{20}$$

This global update is applied to the elements of the best found solution and to some neighbouring elements only. More precisely, if for a given location v_{n+j} the number of ambulances assigned is s, then not only the pheromone value τ_{js} is updated but also the two neighbouring elements τ_{js-1} and τ_{js+1}. The increment $\Delta\tau_{js}$ is given by

$$s = \begin{cases} \Delta\tau_{js'} = 0.8 \text{ if } s' = s \\ \Delta\tau_{js'} = 0.1 \text{ if } s' \in \{s-1, s+1\}. \end{cases} \tag{21}$$

The motivation for including the neighboring solution elements into the update rule is that in our problem, a number of ambulances similar to the one in the best found solution should also lead to good results. We do not know a publication in the ACO literature where the dissemination of pheromone to the neighbour values has already been used. We introduced this technique to increase the robustness of the pheromone update mechanism.

3.1.6 Local Search After an ant has constructed a solution, a local search procedure is applied. In our approach, we use a local search similar to the one embedded in the TS of [11]. For each location $v_{n+j} \in W$, the possible neighbourhood moves consist of moving one ambulance to one of the k nearest locations. In our experiments, we have chosen the same value for k as in [11], namely $k = 5$. As soon as there is an improvement in the objective function, the move is accepted and the local search procedure is restarted.

3.2 Approach 2: Tabu Search

In this section the re-implementation of the Tabu Search algorithm of [11] will be described. A detailed explanation of the tabu search algorithm applied to the ambulance location problem is given in [11].

3.2.1 Initial Solution and Neighbourhood Structure In our implementation, we modified some steps of the original algorithm for the sake of simplicity. Instead of solving a relaxed problem, the initial solution is generated randomly, by placing a random number of ambulances at each potential location. The basic operation in the tabu search algorithm is displacing one ambulance from a vertex $v_{n+j} \in W$ with $z_j \geq 1$ and moving it to $v_{n+j'} \in W$, where $v_{n+j'}$ has to be among the five nearest possible locations of the vertex v_{n+j}.

The tabu search starts from one solution and generates in each iteration a set $N(z)$ of neighbour solutions. A parameter θ_2 is used to specify the number of neighbouring solutions generated. In the generation phase of the neighbouring solutions, some vertex pairs are temporarily declared tabu, but the tabu status of the vertex pairs is erased at the start of the generation of the next neighbor. Among the θ_2 neighbours the best solution is chosen as the incumbent solution.

The ordered pair $(j^{'}, j)$ corresponding to the new incumbent is then set tabu with the tabu duration being a random variable θ_1 chosen in some interval $[\overline{\theta_1}, \underline{\theta_1}]$, where $\overline{\theta_1}$ and $\underline{\theta_1}$ are parameters. As long as a move is set tabu, it can not be reversed, i.e., no ambulance can be moved from $v_{n+j'}$ to v_{n+j}.

The stopping criterion for the TS is a maximum number of θ_3 iterations.

3.2.2 Diversification and Stopping Rule A diversification strategy is applied to enable the search to escape from local optima and to move to different regions in the search space. In the tabu search implementation, another parameter θ_4 is introduced. Whenever there is no improvement during θ_4 iterations, a diversification phase is initiated. In the diversification phase instead of considering the five nearest neighbor vertices, all vertices are considered. The diversification phase is stopped when the objective improves and is applied during up to θ_5 iterations.

4 Example Study: Ambulance Location Planning for Austria

4.1 The Application Instances

Austria is a country of about eight million inhabitants located at the center of Europe. The country is divided into nine provinces. In the eight rural provinces (the 9th province is the capital Vienna), the major part of the emergency and patient transport is organised and executed by the Austrian Red Cross [14]. According to the efficiency report of the Austrian Red Cross of the year 2002, this organization has 460 ambulance bases spread over Austria. The vehicle fleet consisted of 1,952 ambulances in the year 2002. In this year, 2,308,304 trips have been carried out. Thereby, 1,790,018 patients were treated, 226,611 persons thereof needed an emergency transport with qualified ambulance personnel on board and 105,274 patients thereof needed an emergency physicians on board. The remaining transports were executed on a dial-a-ride basis (e.g., transportation of handicapped persons, dialysis patients).

In our computational study, we solve the problem defined in Section 2 for the eight rural provinces (Vorarlberg, Burgenland, The Tyrol, Carinthia, Salzburg, Styria, Upper Austria, and Lower Austria) shaded in grey in Fig. 2. Using ArcGIS version 9, we built a GIS (geoinformation system) providing the essential information for our purpose. In particular, we geocoded the potential ambulance bases. For the population figures, we used the last census data provided in the Arc Austria data collection [17]. Our distance information is based on real road network data provided by Teleatlas [16].

As an example, the province map including population nodes and potential ambulance bases (nodes indexed by numbers) for *Salzburg* is depicted in Figure 3.

4.2 Parameter Values

Both for ACO and TS, we used the following weights for the objective function (10):

$M_1 = 1000000$, $M_2 = 20$, and $M_3 = 10$.

The chosen weighting factors M_1, M_2, M_3 were found after some pretests. In our ACO implementation, the following parameter values were used:

$\rho = 0.1$, $q_0 = 0.9$, $\alpha = \beta = 1$, $popsize = 20$, $niter = 100$.

We tested also with a different value of $\rho = 0.05$ and we found out that the results are slightly better when using a higher evaporation rate $\rho = 0.1$ especially for the larger problems.

In the TS implementation, the parameters were set to the following values:

$$\theta_1 = \theta_2 = 10,\ \theta_3 = p \text{ (number of ambulances)},\ \theta_4 = 50,\ \theta_5 = 10.$$

4.3 Results

In Table 1, we list the characteristic parameters of the eight provinces with respect to the given problem.

For the two smallest of the eight provinces (Vorarlberg and Burgenland), we were able to solve our modified problem *exactly*, such that we could use the results as yardsticks for the outcome of the two heuristics. For the six other provinces, problem size prevents the computation of exact solutions in reasonable time, such that the corresponding optima are not known; here, all we can do is to compare the ACO results with the TS results. Table 2 provides the best found solution for Salzburg.

We use real world data for the development of our algorithms. In order to realize this solution in the real world additional constraints and dynamic aspects must be taken into account (e.g. age of the population in certain regions, places with higher probabilities of accidents, tourist regions, industrial regions,...) The consideration of this additional aspects is part of further research and goes beyond the scope of this paper.

For each province, ten runs of ACO and ten runs of TS have been carried out to deal with the randomised character of the two algorithms. In Table 3, the achieved values of the objective function $F(z)$ are presented. Both for ACO and for TS, the corresponding best and worst values as well as the average over the ten runs have been indicated. As can be seen, for both Vorarlberg and Burgenland, both heuristics produced the optimal solution in each run.

A comparison between ACO and TS shows that ACO produces better solutions than TS in three cases, and TS produces better solutions than ACO in three other cases. In the remaining two cases, the solutions produced by the two heuristics are equally good since they are, as stated above, already optimal. It can be observed that TS outperforms ACO in the *largest* of our test instances (Upper Austria, Lower Austria). The number eight of test instances seems to be too small, however, to draw general conclusions from this observation.

In Table 4, we present the achieved *double coverage* values, expressed in percent of the population, i.e., the values $f(z)/\left(\sum_{i=1}^{n}\lambda_i\right)$. Note that $f(z)$ resp. $f(z)/\left(\sum_{i=1}^{n}\lambda_i\right)$ is not the true objective function, because it does not include the penalties $f_1(z)$ to $f_3(z)$.

Computation times are indicated in Table 5. As it can be observed, the computation times of ACO are about 3 to 4 times higher than those of TS for the smaller instances. For the three largest instances, this factor increases to between 10 and 15. For the problem instances where ACO provides better results we increased the runtime of the TS algorithm (without modifying the parameter settings for the TS). Even if we apply the TS for the same runtime than ACO, the TS cannot outperform the ACO algorithm.

The reason for the better performance of the TS for the larger problems is that TS is an improvement procedure and it improves one solution iteratively. The ACO algorithm always constructs a population of new solutions - this process consumes a lot of runtime and therefore not so many new solutions can be constructed during the search. Recently, a first work was published where also the ants do not always construct new solutions and just improve solutions [1]. Maybe this is a first step towards an ACO algorithm which does not consume so much runtime in generating or constructing new solutions.

5 Conclusions

We have solved an ambulance location problem with a modified double-coverage objective function under single-coverage constraints for eight provinces of Austria. The optimization model generalises the model developed by Gendreau, Laporte and Semet [11] by requiring a more balanced distribution of ambulances with respect to the number of inhabitants served by one ambulance. As solution algorithms, we used two metaheuristic approaches: A variant of Ant Colony Optimization specifically developed in this paper for the considered location problem, and the Tabu Search algorithm from [11]. The results showed no consistent superiority in solution quality of one of the approaches over the other. However, Tabu Search always consumed less computation time.

Further research will be directed to more elaborate models for measuring the service level achieved by certain proportions between number of inhabitants and number of ambulances. To increase realism, it may be necessary to include stochastic influences and queuing into the model, as it has been done, e.g., in [2] or in [13]. A further interesting extension would be to treat the terms in our objective function (10) as single, separate objectives and to determine Pareto-optimal solutions of the corresponding multiobjective optimization problem.

Acknowledgement. We want to thank Gilbert Laporte for some valuable comments on the design of the Tabu Search algorithm. We are grateful to Michael Niemeck from the Austrian Red Cross for helpful suggestions concerning the real world situation. Financial support from the Oesterreichische Nationalbank under grant #11187 is gratefully acknowledged.

References

1. Acan, A. (2004), "An External Memory implementation in Ant Colony Optimization", Dorigo, M., Birattari, M., Blum, Ch. et al. (eds.), Proceedings of the 4th International Workshop, ANTS 2004, Brussels, Belgium, September, 5-8, 2004. Lecture Notes in Computer Science 3172. pp. 73-82.
2. Ball, M.O., Lin, F.L. (1993), "A Reliability Model Applied to Emergency Service Vehicle Location", *Operations Research* 41, pp. 18-36.

3. Brotcorne, L., Laporte, G., Semet, F. (2003), “Ambulance Location and Relocation Models”, *European Journal of Operational Research* 147, pp. 451-463.
4. Church, R.L., ReVelle, C. (1974), “The Maximal Covering Location Problem”, *Papers of the Regional Science Association* 32, pp. 101-118.
5. Colorni, A., Dorigo, M. and Maniezzo, V. (1992), “Distributed Optimization by Ant Colonies”, in: *Toward a Practice of Autonomous Systems: Proceedings of The First European Conference on Artificial Life*, The MIT Press.
6. Daskin, M. (1983), “A Maximum Expected Covering Location Model: Formulation, Properties and Heuristic Solution”, *Transportation Science* 17, pp. 48–70 .
7. Doerner, K., Gutjahr, W.J., Hartl, R., Strauss, C. and Stummer, C. (2005), “Nature Inspired Metaheuristics in Multiobjective Activity Crashing”, Technical report.
8. Doerner, K., Focke, A., Gutjahr, W.J. (2005), “Multicriteria Tour Planning for Mobile Healthcare Facilities in a Developing Country”, Technical report.
9. Dorigo, M., Maniezzo, V., Colorni, A. (1996), “The Ant System: Optimization by a Colony of Cooperating Agents”, *IEEE Transactions on Systems, Man and Cybernetics* 26, pp. 29–41 .
10. Dorigo, M., Stuetzle, Th. (2004), *Ant Colony Optimization*, MIT Press, Cambridge, Massachusetts.
11. Gendreau, M., Laporte, G., Semet, F. (1997), “Solving an Ambulance Location Model by Tabu Search”, *Location Science* 5, pp. 75-85.
12. Gutjahr, W.J. (2002), “ACO Algorithms with Guaranteed Convergence to the Optimal Solution”, *Information Processing Letters* 82, pp. 145-153.
13. Marianov, V., ReVelle, C. (1996), “The Queueing Maximal Availability Location Problem: A Model for the Siting of Emergency Vehicles”, *European*

Journal of Operational Research 93, pp. 110–120.

14. Efficiency Report of the Austrian Red Cross (www.roteskreuz.at/376.html), 2002. accessed April 1st, 2005.
15. Scaparra, M. P. and Scutella, M.G. (2001) “Facilities, Locations, Customers: Building Blocks of Location Models. A Survey”, Technical Report.
16. http://www.teleatlas.com
17. http://www.wigeogis.at

Procedure ACOLoc

```
create popsize ants;
initialise pheromone matrix;
do for niter iterations {
    for ant = 1 to popsize {
        determine the visiting sequence of locations v_i ∈ W randomly
        by using a uniform distribution;
        for i = 1 to m
            select a number of vehicles c_i using formula (18);
        apply local pheromone update using formula (19);
        apply local search to the ant's solution;
    }
    from the solutions just found by the popsize ants
    determine the best solutions z*;
    do pheromone update on z* using formula (20);
}
```

Fig 1. Pseudocode of the ACO algorithm

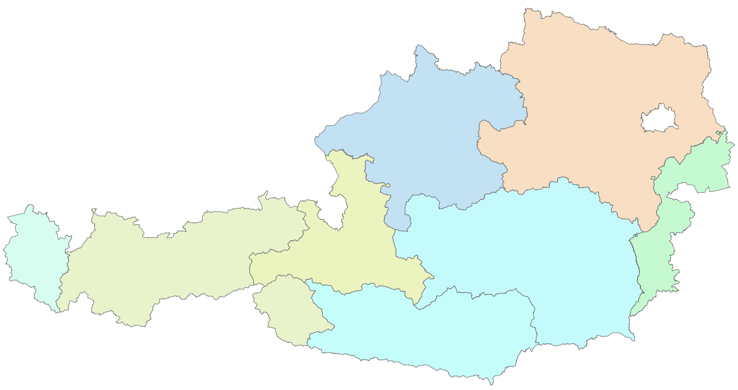

Figure 2. Map of the nine provinces in Austria.

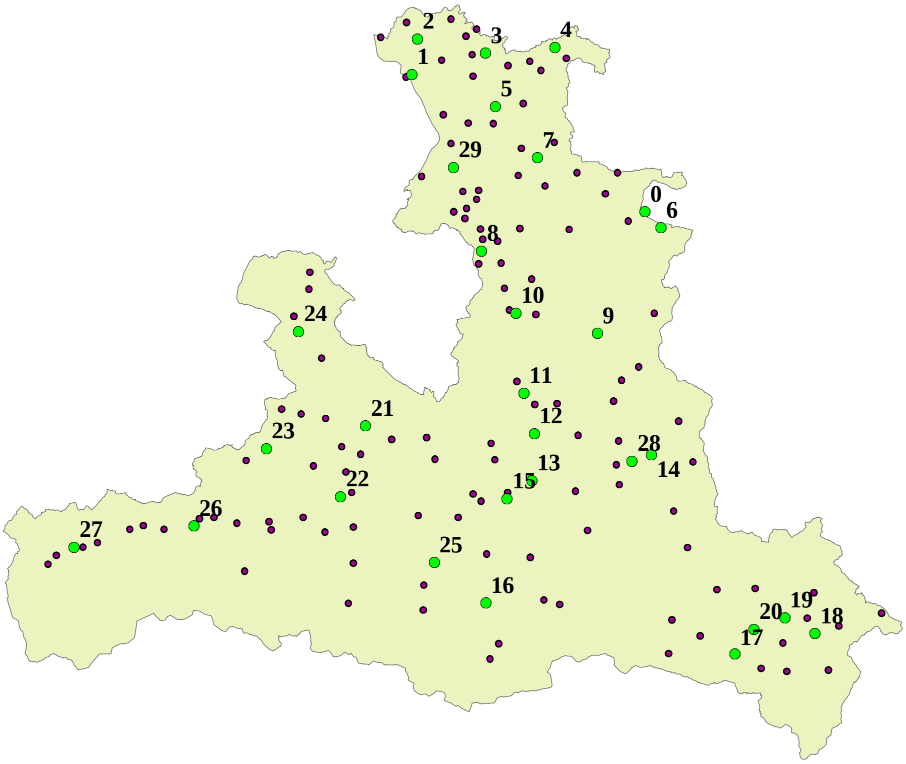

Figure 3. Map of the province Salzburg.

Province	No. demand points	No. pot. locations	No. Ambul.	No. inhabitants	No. inhabitants /No. ambulances	r	R
Vorarlberg	105	7	11	594,000	54,000	20	57
Salzburg	152	30	40	504,000	12,600	15	39
Lower Austria	723	137	200	2,678,000	13,390	16	35
Carinthia	319	22	35	990,000	28,300	20	48
Upper Austria	443	86	120	1,902,000	15,000	15	36
Burgenland	189	10	15	634,000	30,000	20	39
Styria	451	93	150	1,486,000	10,000	22	37
The Tyrol	269	43	60	826,000	19,200	15	32

Table 1 Problem size and characteristics of the different provinces.

Salzburg

ID	Location Name	No. of Ambulances
0	St. Wolfgang	0
1	Oberndorf	1
2	Flachgau-Nord	1
3	Flachgau-Mitte	1
4	Strasswalchen	0
5	Seekirchen	1
6	Strobl	1
7	Flachgau-Ost	3
8	Tennengau	1
9	Lammertal	1
10	Golling	2
11	Werfen	1
12	Bischofshofen	2
13	Pongau	2
14	Radstadt	1
15	Schwarzach	1
16	Gastein	4
17	St. Michael	2
18	Lungau	1
19	Mariapfarr	1
20	Mauterndorf	1
21	Saalfelden	2
22	Pinzgau	2
23	Saalbach	0
24	St. Martin b. Lof	1
25	Rauris	1
26	Mittersil	2
27	Wald im Pinzgau	1
28	Altenmarkt	1
29	Salzburg	2

Table 2 Best solution found for the province of Salzburg.

.	ACO	avg. $F(s)$	best	worst
1	Vorarlberg	435,491	435,491	435,491
2	Salzburg	155,996	155,996	155,996
3	Lower Austria	1,304,593	1,337,280	1,269,950
4	Carinthia	228,742	228,742	228,742
5	Upper Austria	1,040,429	1,051,270	1,029,610
6	Burgenland	111,163	111,163	111,163
7	Styria	790,721	797,996	784,060
8	The Tyrol	405,996	405,996	400,080

.	TS	avg. $F(s)$	best	worst	optimum
1	Vorarlberg	435,491	435,491	435,491	435,491
2	Salzburg	155,117	155,996	151,996	n.a.
3	Lower Austria	1,345,833	1,353,540	1,340,320	n.a.
4	Carinthia	228,250	228,740	227,718	n.a.
5	Upper Austria	1,039,200	1,051,350	1,021,920	n.a.
6	Burgenland	111,163	111,163	109,496	111,163
7	Styria	793,819	797,996	788,694	n.a.
8	The Tyrol	398,862	402,697	396,615	n.a.

Table 3 Objective function values $F(z)$.

.	ACO	avg. $f(s)$	avg. %	best	worst
1	Vorarlberg	444,000	74.75	74.75	74.75
2	Salzburg	156,000	30.95	30.95	30.95
3	Lower Austria	1,331,000	49.70	50.56	48.84
4	Carinthia	238,000	24.04	24.04	24.04
5	Upper Austria	1,048,800	55.14	55.52	54.57
6	Burgenland	178,000	28.08	28.08	28.08
7	Styria	794,600	53.47	53.70	53.16
8	The Tyrol	406,000	49.15	49.15	49.15

.	TS	avg. $f(s)$	avg. %	best	worst	optimum	in %
1	Vorarlberg	444,000	74.75	74.75	74.75	444,000	74.75%
2	Salzburg	155,200	30.79	30.95	30.16	n.a.	n.a.
3	Lower Austria	1,361,715	50.85	50.93	50.63	n.a.	n.a.
4	Carinthia	238,000	24.04	24.04	24.04	n.a.	n.a.
5	Upper Austria	1,049,400	55.17	55.73	54.26	n.a.	n.a.
6	Burgenland	178,000	28.08	28.08	28.08	178,000	28.08%
7	Styria	797,400	53.66	53.70	53.30	n.a.	n.a.
8	The Tyrol	406,000	49.15	49.15	49.15	n.a.	n.a.

Table 4 Double coverage.

	Comp. Time	ACO	TS
1	Vorarlberg	5-8 sec	1-2 sec
2	Salzburg	80 sec	15 sec
3	Lower Austria	3.5 h	20 min
4	Carinthia	140 sek	20 sec
5	Upper Austria	1 h	6 min
6	Burgenland	18 sec	10 sec
7	Styria	1.33 h	6 min
8	The Tyrol	6.5 min	2.5 min

Table 5 Computation time.