Multiple Ant Colony Optimization for a Rich Vehicle Routing Problem: A Case Study

Paola Pellegrini, Daniela Favaretto, and Elena Moretti

Department of Applied Mathematics, University Ca' Foscari of Venice
Dorsoduro 3825/E, I–30123 Venice, Italy
paolap@pellegrini.it, favaret@unive.it, emoretti@unive.it

Abstract. Starting from a case study, a rich vehicle routing problem is analyzed. It is characterized by multiple time windows, heterogeneous fleet, maximum duration, and multiple visits. Two variants of Ant Colony Optimization are proposed in a multiple colonies framework. Two algorithms are tested, giving results that appear satisfactory with respect to the ones achieved by the firm.

Keywords: ant colony optimization, vehicle routing, case study.

1 Introduction

The analysis at the basis of this paper is motivated by the need of solving a case study, which can be modeled as a rich vehicle routing problem.

The vehicle routing problem (VRP) consists in the determination of the optimal set of routes to serve a given set of customers using a fixed fleet of vehicles. In recent years, thanks to the increasing efficiency of heuristic and metaheuristic approaches and the availability of a larger computing power, the interest has been shifted to variants which combine various types of constraints, such as: time windows in which the service must be performed, the requirement of inserting in the same tours both pickup and delivery of goods, the availability of different types of vehicles, the presence of more than one depot. These variants are grouped under the denomination of rich VRP.

The rich vehicle routing problem studied in this paper is characterized by: the presence of multiple time windows, the availability of different types of vehicles, the requirement of multiple visits to some customers and the limitation of the duration of each subtour. Moreover, two objectives are to be considered in hierarchic order. The first one is the minimization of the number of vehicles used, and the second one is the minimization of the total time required by the subtours.

We use the metaheuristic approach known as Ant Colony Optimization [1] to tackle this problem. We consider two of its most successful variants: Ant Colony System [2] and $\mathcal{MAX}$–$\mathcal{MIN}$ Ant System [3]. They are both studied in a multiple colonies framework. The respective results are compared on a set of instances generated on the basis of real ones. The Randomized Nearest Neighbor heuristic [4] and a Tabu Search algorithm are used as reference elements.

B. Apolloni et al. (Eds.): KES 2007/ WIRN 2007, Part II, LNAI 4693, pp. 627–634, 2007.

The algorithm which performs the best on the generated instances, is then used for a comparison with the solutions proposed by the firm considered.

First of all, the case study is presented. Secondly, the resulting rich vehicle routing problem is proposed. Then the two Ant Colony Optimization algorithms considered and Tabu Search approach are described. A following section describes the instances used for the analysis and another one reports the computational results obtained. Finally some conclusions are drawn.

2 The Case Study

The case study at the basis of this analysis is concerned to an Italian firm. Its major activity consists in the delivery of a wide number of food products to restaurants and retailers in the North-East region of the country. This task is accomplished by external suppliers of vehicles. The cost for completing the services consists in a fixed amount related to each vehicle used and in a variable one due to the working hours of the drivers. The firm decides the set of customers to be assigned to each vehicle and the sequence according to which the customers must be visited. The fee owed to a driver is much lower for the first eight hours than for the following ones, and then the firm imposes tours of limited duration.

This structure allows to consider as virtually unlimited the dimension of the fleet available. The usable vehicles are of two kinds. They differ for what concerns the external dimensions and the load and unload system. The use of the a specific type may be necessary for serving a precise customer.

As it is widespread practise, the firm allows customers to choose when to receive the delivery. To this aim, they can indicate at most three time intervals for each of the five working days of a week. This gives a total of at most fifteen time windows per customer. Moreover, each customer can require to be served more than once a week. When this is the case, the firm must take care of placing the visits in non-consecutive days unless the customer himself explicitly requires consecutive visits.

The aim of the firm is the minimization of the cost of completing all the deliveries. The fix cost, related to the utilization of the vehicles, has a stronger impact than the variable one concerning the total time needed. This makes a set of few tours requiring a certain time preferable to another set including more tours and implying a smaller amount of hours. The general aim then can be split in two objectives which have to be pursued hierarchically: the first is the minimization of the number of vehicles required to complete the services, the second is the minimization of the total time needed.

The procedure currently applied by the firm to establish the tours consists first of all in splitting the customers in groups of seventy to eighty, and then in deciding the assignments to the different vehicles according to a weekly time horizon. Tours are scheduled on the basis of the experience and the knowledge of the territory of the logistic manager. In agreement with the firm, the aim of the research presented in this paper is proposing an algorithm for solving this second step, considering the first clusterization as non-modifiable.

3 The Rich Vehicle Routing Problem

To the best of our knowledge, the rich vehicle routing problem that is object of this study has not been considered in the literature. The element which differentiates the most the current variant from the ones that can be found in the literature, is the objective function. Often, only one objective is considered. Moreover, the quantity which is typically minimized is the total distance (or traveling time), while in this study we must take into account also the waiting time.

Let us focus on the most particular constraints of our problem. A first type of constraint is related to the requirement of multiple visits to a single customer. These visits must be scheduled at a suitable time distance, as discussed in the following. The time horizon considered is split in disjoint subperiods, which represent different days. For each day, customers may indicate one or more time windows in which to be served. Each subtour must belong to only one of the subperiods and its total time length must not exceed a fixed duration. Furthermore, an heterogeneous fleet is available, and each customer may require to be served by a specific type of vehicle. A specific type of vehicle, then, must be associated to each subtour.

A mathematical formulation of a very similar model can be found in [5].

4 Multiple Ant Colony System and Multiple $\mathcal{MAX}$–$\mathcal{MIN}$ Ant System

Ant Colony Optimization (ACO) is a metaheuristic based on the foraging behavior of ants. A complete analysis of this metaheuristic can be found in [1]. Many ACO algorithms have been proposed in the literature. The two most successful variants, when dealing with routing problems are recognized to be Ant Colony System (ACS) first proposed in [6] and $\mathcal{MAX}$–$\mathcal{MIN}$ Ant System ($\mathcal{MM}$AS) first introduced in [3]. In this study, the *swap* local search is combined with the algorithm. This procedure is well known in the literature and it was first proposed by Lin in 1965. It offers the advantage of not being very expensive in terms of computational time although it offers satisfactory results. In our case this element is crucial since the addition of many constraints could make other typical local search procedures very time consuming.

In order to apply the two algorithms to the rich VRP considered, some elements must be clarified.

For taking into account the differences among vehicles, the following procedure is applied. First of all, one type of vehicle $\hat{k}$ is fixed as default. Then the insertion of the customers begins considering the capacity constraint imposed by type $\hat{k}$, and without taking into account the type of vehicle possibly required by the customers. When the first customer requiring a specific type of vehicle is inserted, this type is considered for the subtour. The following customers will be feasible either if they do not require a specific class of vehicles or if they require the class already established. When considering customers for the insertion while the vehicle is still the default one, if the cumulated demand is larger than the capacity of some type of vehicle ($\tilde{k}$), then all the customers requiring $\tilde{k}$ are set unfeasible.

Periodic constraints must be satisfied. Before starting all the procedures, the set of customers is analyzed. Each of those requiring more than one visit is duplicated once for each visit required. To each dummy customer the edges and some time windows of the original one are assigned. If the partition of the time windows allows a suitable separation between the services, it is recorded that the visits must happen with at least a certain distance. This distance depends from the number of visits required. For a formal description of this procedure we refer the reader to [5].

As far as the multiple time windows are concerned, the procedure used is quite straightforward. It consists in choosing dynamically the one to be considered. This time window corresponds to the first one closing after the instant in which the customer can be reached, if such time window exists. The number and duration of the not yet closed time windows is nonetheless taken into account. In particular, the choice is biased in favor of customers that would be the hardest to serve in the future, considering all the time windows they indicate. This is done by introducing a heuristic measure proposed in [5] in the random proportional rule (pseudo-random proportional rule for ant colony system) [1].

The constraint on the duration of each subtour is treated by stating that, if visiting customer j and going back to the depot implies that the current subtour lasts more than the available time, then j is unfeasible.

Finally, the algorithms must operate considering two objectives in hierarchical order. They represent the minimization of the number of vehicles used, and the minimization of the time required to complete the tour. To this aim, we follow the approach presented in [7]: two kinds of colonies with different specializations are exploited. This system has been developed to tackle a multi-objective vehicle routing problem with time windows using Ant Colony System and has never been implemented for other ACO variants, despite reaching very encouraging results.

The two algorithms obtained following these steps will be referred as Multiple Ant Colony System (M-ACS) and Multiple $\mathcal{MAX}$–$\mathcal{MIN}$ Ant System (M-$\mathcal{MM}$AS).

5 Tabu Search

For evaluating the performance of the M-ACS and M-$\mathcal{MM}$AS, we consider as reference element a Tabu Search algorithm. Tabu Search [8] is one of the most successful metaheuristics for the vehicle routing problem with single time windows [9]. For this reason it is considered as a reference point for the algorithms proposed here. In the algorithm, the starting solution is generated with the Nearest Neighbor heuristic and its neighborhood is explored via the *swap* local search procedure. Following [10], a *variable neighborhood* is considered: for any current solution, instead of considering the whole neighborhood, we choose a subset of it according to a given probability distribution.

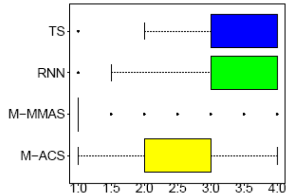


Fig. 1. Ranking of the performance of the algorithms

6 Experimental Results

In order to choose which algorithm to suggest to the firm, we compared the performance of Multiple Ant Colony System, Multiple $\mathcal{MAX}$–$\mathcal{MIN}$ Ant System, Tabu Search and Randomized Nearest Neighbor (RNN) [4] on a set of instances generated on the basis of the real ones.

As reported when the case study has been presented, we are interested in sets of customers with cardinality of the order of 70 to 80 elements, as for the subsets of customers considered by the firm. The area actually served by the firm is used as a reference for the localization of the customers. We use for the analysis a set of instances generated in order to mimic the real instances proposed by the firm [4].

Experiments on Generated Instances

The experiments are run on a cluster of 32 AMD Opteron$^{\text{TM}}$ 244. The executables have been generated from C++ source. The code is publicly available on the web page www.paola.pellegrini.it. The common practice used for dealing with metaheuristics is followed [11,12,13]. A set of 400 instances is used for the experiments.

The behavior of the Deterministic Nearest Neighbor (DNN) is considered as reference. The element that first emerges is the dominance of the M-$\mathcal{MM}$AS over the other algorithms. In Figure 1 the ranking of the different approaches is reported. As it appears evident in the strong majority of the cases the M-$\mathcal{MM}$AS outperforms the other approaches. The difference appears statistically significant according to the Friedman two-way analysis of variance by ranks. This better behavior is observable in Table 1, in which the number of instances with an improvement with respect to DNN, and the average percentage of these improvement, are presented for each algorithm. The improvement with respect to the first objective (minimum number of vehicles) is reported in the first line. The one concerning the minimum traveling and waiting time is given in the second line. Owing to the priority of the minimization of the number of vehicles, if a solution is preferable in this sense the travel time is not considered. The second objective is checked when no difference is detectable about the first. As it can be seen, in general M-$\mathcal{MM}$AS and M-ACS behave in a comparable way. They improve significantly the number of vehicles used in the best solution

Table 1. Number of instances in which any improvement with respect to the DNN has been found and average percentage improvement

M-ACS		M-$\mathcal{MM}$AS		RNN		TS	
veh.	%	veh.	%	veh.	%	veh.	%
395	21.36	375	25.06	213	12.66	321	12.84
time	%	time	%	time	%	time	%
0	0.00	0	0.00	97	11.91	42	8.84

found and do not make a real difference with respect to the total travel time when they are not able to make a difference in terms of the first objective. It can be observed, moreover, that even if the number of instances in which the number of vehicles decreases with respect to the DNN is very similar for the two algorithms, the average value of this difference is much higher when considering M-$\mathcal{MM}$AS. A completely different behavior is observable for RNN, where the number of vehicles is quite seldom improved (in about 50% of the cases). When this happens the difference is less stressed than for the ACO algorithms. On the other hand the total travel time is diminished much more often. Finally, TS places itself between the ACO algorithms and RNN. The number of instances in which the number of vehicles is improved is close to the one detected for ACO algorithms, and once it is even bigger. Nonetheless, the average improvement is always quite lower. The value referred to the total travel time in this case is comparable to the one concerning RNN. The differences reported in Table 1 concerning the number of vehicles, are significant according to the Wilcoxon test.

Experiments on Real Instances

Let us try to compare the approaches presented with the one currently applied by the firm object of the case study. It is not very easy to analyze the differences, since following the tours provided by the firm about 20% of the time windows are violated. In any case, the ACO algorithms find feasible solutions which imply in average the same number of vehicles currently used, even if the total travel time needed is bigger. TS is the third algorithm, followed by RNN and DNN, all supplying tours requiring more vehicles than those currently used but clearly not violating any constraint. For giving a clearer idea of the different solutions, let us consider a representative instance. It is graphically shown in Figure 2 and the characteristics of the customers follow the trends reported in the description of the instances generated. The tours proposed by the firm and the ones found by $\mathcal{MAX}$–$\mathcal{MIN}$ Ant System are graphically shown in Figure 3. The customers whose time windows are violated are represented by black bullets. As it can be seen, in the tour proposed by the firm there is a sort of equilibrium in the number of customers served by each vehicle. In the solution found by $\mathcal{MAX}$–$\mathcal{MIN}$ Ant System this element is not detectable. The disequilibrium may appear as a weakness of the set of tours proposed by the ACO algorithm. On the other hand, the fact that only a few customers are inserted in the last two tours, implies that relaxing only a few constraints, such as some time windows, it might be

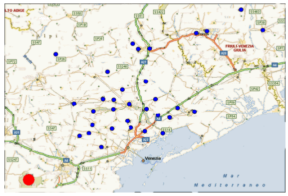
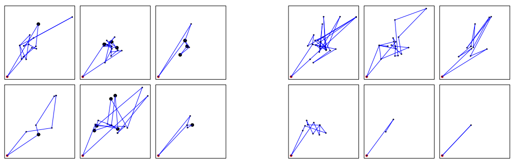


Fig. 2. Graphical representation of the instance considered



(a) (b)

Fig. 3. Solution proposed by the firm and by $\mathcal{MAX}$–$\mathcal{MIN}$ Ant System

possible to eliminate them, and then to complete the services with two vehicles less. Moreover, by slightly changing the clusters which currently are considered fixed under suggestion of the firm – as previously explained – it would probably be possible to find more efficient solutions.

7 Conclusion

Two Multiple-ACO algorithms are presented for dealing with a case study which can be modeled as a rich vehicle routing problem. Its characteristics are: multiple hierarchically ordered objectives, multiple time windows, heterogeneous fleet of vehicles, periodic constraints and limited duration of the subtours. As reference elements the Deterministic and Randomized Nearest Neighbor heuristics and a Tabu Search algorithms are considered. The ACO algorithms proposed perform significantly better than these alternatives, both on instances generated for the theoretical analysis and on those available for the case study. The results of this study are currently being exploited by the firm which proposed the analysis.

Acknowledgements. The authors would like to thank Mauro Birattari for the useful advice during the preparation of this paper. They also thanks IRIDIA, Université Libre de Bruxelles, Belgium, for allowing the use of computing power.

References

1. Dorigo, M., Stützle, T.: Ant Colony Optimization. MIT Press, Cambridge (2004)
2. Dorigo, M., Gambardella, L.M.: Ant colonies for the traveling salesman problem. ByoSystems 43, 73–81 (1997)

3. Stützle, T., Hoos, H.: The $\mathcal{MAX}$–$\mathcal{MIN}$ Ant System ant system and local search for the traveling salesman problem. In: Angeline, P. (ed.) Proceedings of the IEEE International Conference on Evolutionary Computation, Indianapolis, USA, pp. 308–313. Springer, Heidelberg (1997)
4. Pellegrini, P.: Application of two nearest neighbor approaches to a rich vehicle routing problem. Technical Report TR/IRIDIA/2005-15, Iridia, Université Libre de Bruxelles, Brussels, Belgium (2005)
5. Favaretto, D., Moretti, E., Pellegrini, P.: Ant colony system for a vrp with multiple time windows and multiple visits. Journal of Interdisciplinary Mathematics (to appear, 2006), Available on `http://www.paola.pellegrini.it`
6. Dorigo, M., Gambardella, L.M.: Ant colony system: A cooperative learning approach to the traveling salesman problem. IEEE Transaction on Evolutionary Computation 1, 53–66 (1997)
7. Gambardella, L.M., Taillard, E., Agazzi, G.: MACS-VRPTW: A multiple ant colony system for vehicle routing problem with time windows. In: Corne, D., Dorigo, M., Glover, F. (eds.) New Ideas in Optimization, pp. 63–76. McGraw-Hill, London (1999)
8. Glover, F.: Future paths for integer programming and links to artificial intelligence. Computers & Operations Research 13, 533–549 (1986)
9. Cordeau, J.F., Laporte, G., Mercier, A.: A unified tabu search heuristic for vehicle routing problem with time windows. Journal of the Operational Research Society 59, 928–936 (2001)
10. Bianchi, L., Birattari, M., Chiarandini, M., Manfrin, M., Mastrolilli, M., Paquete, L., Rossi-Doria, O., Schiavinotto, T.: Hybrid metaheuristics for the vehicle routing problem with stochastic demand. Journal of Mathematical Modelling and Algorithms 5(1), 91–110 (2006)
11. Birattari, M.: On the estimation of the expected performance of a metaheuristic on a class of instances. How many instances, how many runs? Technical Report TR/IRIDIA/2004-01, Université Libre de Bruxelles (2004)
12. Birattari, M., Stützle, T., Paquete, L., Varrentrapp, K.: A racing algorithm for configuring metaheuristics. In: Langdon, W. (ed.) GECCO 2002: Proceedings of the Genetic and Evolutionary Computation Conference, pp. 11–18. Morgan Kaufmann Publishers, San Francisco, CA (2002)
13. Birattari, M., Zlochin, M., Dorigo, M.: Towards a theory of practice in metaheuristics design: A machine learning perspective. Theoretical Informatics and Applications 40(2), 353–369 (2006)