Cooperative Ant Colonies for Optimizing Resource Allocation in Transportation

Karl Doerner, Richard F. Hartl, Marc Reimann [1]

Abstract

In this paper we propose an ACO approach, where two colonies of ants aim to optimize total costs in a transportation network. This main objective consists of two sub goals, namely fleet size minimization and minimization of the vehicle movement costs, which are conflicting for some regions of the solution space. Thus, our two ant colonies optimize one of these subgoals each and communicate information concerning solution quality. Our results show the potential of the proposed method.

1 Introduction

In the last decade, a new meta-heuristic called Ant Colony Optimization (ACO) has attracted increasing attention, as a tool to solve various hard combinatorial optimization problems (cf. e.g. [1], [2], [4], [8], [12], [9]). It is based on research done in the early nineties by Dorigo et al. (see e. g. [3], [6], [7]) on the Ant System, which was inspired by the behaviour of real ant colonies searching for food. Information concerning the quality of food sources is communicated between the

[1]Department of Production and Operations Management, Institute for Management Science, University of Vienna, Bruennerstrasse 72, 1210 Vienna, Austria, Tel.: ++43 1 4277-38096, Fax: ++43 1 4277-38094, e-mail: marc.reimann@univie.ac.at

members of the colony via an aromatic essence called pheromone. Over time this information will lead to the reinforcement of some paths, which lead to rich food sources, while other paths will not be used anymore.

In the context of combinatorial optimization problems this mechanism is implemented as an adaptive memory which, together with a local heuristic function called visibility, guides the search of the artificial ants through the solution space. Thus, the articficial ants base their decisions on their own rule of thumb and on the experience of the colony as a whole. The objective values correspond to the quality of the discovered food. A convergence proof for a generalized Ant System Algorithm is provided in [10].

In order to be able to succesfully implement such an ACO algorithm for a given combinatorial optimization problem, problem specific knowledge is necessary to identify an appropriate rule of thumb to guide the search of the ants to promising regions of the search space. However, for most problems more than one good heuristic exists, and the quality of each one generally depends on the current problem constellation. Furthermore, for problems with multiple goals available heuristics which seek to optimize the objective function with respect to one goal can perform rather poor with respect to some other goals. Thus, such problems are normally solved using either a lexicographic approach or an objective function, which sums up the values associated with each goal. The former approach is based on a ranking of the goals with respect to their importance, in the latter approach each goal can be assigned a weight before the sum is taken.

The aim of this paper is to overcome these problems and develop a method which finds comprehensive solutions for problems with multiple objectives. We propose an algorithm, where two colonies of ants aim to optimize total costs in a transportation network. These costs consist of fixed costs associated with the fleet size and variable vehicle movement costs. In general, minimal fleet sizes will not cause minimal vehicle movement costs, these two goals are rather conflicting. Thus, our two ant colonies optimize one goal each and communicate information about good solutions in order to enhance the general solution quality. However, due to the size of the costs the main goal is to minimize the fleet size required. Thus, in our approach we have a master population which optimizes the fleet size. This 'master' population is supported by a 'slave' population which optimizes empty vehicle movements and communicates outstanding solutions to the 'master' population.

The remainder of this paper is organized as follows. In section 2 we describe the problem we consider. Our new approach for handling multiple objectives is proposed in section 3. In section 4 we present our numerical results. We close in section 5 with some final remarks and an outlook on future research.

2 Description of the problem

The problem considered in this paper represents a typical scenario logistics service providers are faced with. They have to satisfy customer orders, which require the delivery of goods between pickup and delivery locations. In general, if these orders are of small size and distances are large, they are not transported directly from their source to their destination but via the locations of the service provider. Such a situation is depicted in Figure 1. An order which has to be delivered from customer a to customer b is first shipped to the distribution center i associated with customer a. There, together with other orders requiring transportation to the same region, it is consolidated to a full truckload and delivered to the receiving distribution center j, from where it is finally delivered to customer b. While the local transportation is generally performed with small trucks, such a problem is treated in [11], the long distance movements between distribution centers are performed by larger trucks. It is obvious that the utilization of these large trucks is very important for the service provider. Thus, empty vehicle movements have to be minimized. Apart from that, the necessary fleet size should be as small as possible to keep the fixed costs low.

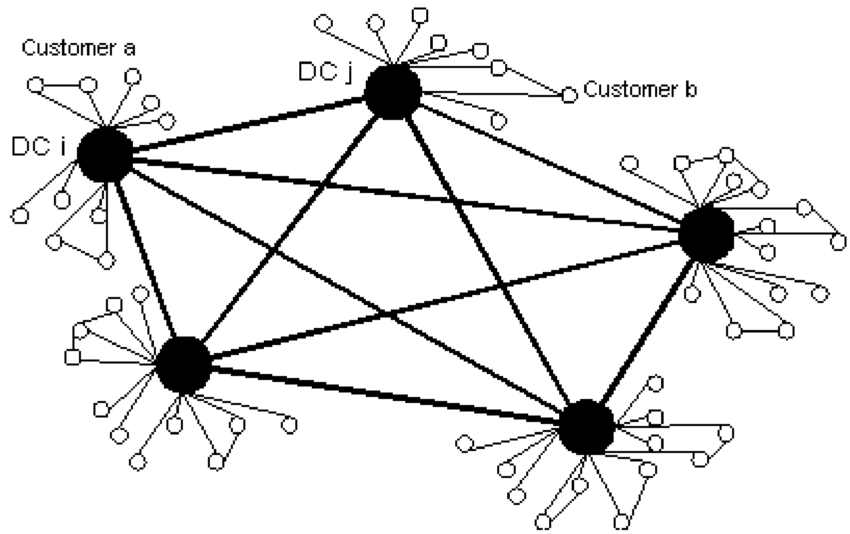

Figure 1: Structure of the distribution network

In this paper, we aim to optimize the total costs in this long distance distribution center network (depicted with bold lines in the figure). In order to achieve this, fleet sizes as well as empty vehicle movements have to be minimized. Note, that these two goals may not be equally important. In a situation, where a fleet of vehicles and a number of drivers are available fleet size may be less important, especially if unused trucks or drivers could not be utilized for other services. However, in our case we look at a situation, utilization of trucks or drivers for alternate services is possible, and thus fleet size minimization is the main goal. A number of constraints have to be considered in order to obtain feasible solutions.

These are:

- All order information is known at the time of planning.
- The service provider operates a fleet of homogeneous vehicles.

- Each vehicle can be used repeatedly within the planning horizon.
- The vehicles have to return to their home base after a given time period.
- Customer time windows have to be respected.
- Transportation orders in the long distance network are consolidated to full truckloads.

Given these constraints we will now propose our new approach developed for this multi-objective problem.

3 Cooperative Ant Colonies to handle multiple objectives

In our algorithm two objectives are optimized simultaneously by coordinating the activities of two ant colonies. The ants of the larger master population optimize the main objective function. The ants of the smaller slave population optimize a minor goal and 'inject' their knowledge into the pheromone information of the main population. The two ant colonies are coordinated by the procedure `CooperativeACO`.

The procedure `CooperativeACO`, described in Table 1 initializes the two ant colonies (master population and slave population), handles the communication of the two populations via one global pheromone information and controls the termination of the algorithm. In detail, for a number of *max_Iterations* the procedure `ACO` is executed for the 'master' population as well as the 'slave' population. Both algorithms are called with the parameter settings for the priority rule (η^{master}, η^{slave}), population size (Γ^{master}, Γ^{slave}), and the number of best ants (Λ^{master}, Λ^{slave}) for the pheromone update. After the run of the two `ACO` Procedures the `PheromoneInjection` for the master population is executed by using the results

Table 1: The CooperativeACO procedure

```
procedure CooperativeACO {
  for i := 1 to max_Iterations {
    ACO (η^master; Γ^master; Λ^master);
    solution_vectors_slave_pop:=
    ACO (η^slave; Γ^slave; Λ^slave);
    PheromoneInjection to the pheromone information of the master population
    using solution_vectors_slave_pop, pheromone information of the 'master' population
    and formula (6);
  }
}
```

(*solution_vectors_slave_pop*), produced by the `ACO` procedure with the parameter settings for the 'slave' population. The resulting pheromone information is replicated in the local pheromone information of the 'master' population.

Let us now turn to a detailed description of the ACO procedure (subsection 3.1), where we briefly describe the two basic ACO phases, namely the construction of a feasible solution and the trail update for the global pheromone information (in section 3.2).

3.1 The Ant Colony Algorithm

The master and the slave population use the same ACO algorithm to construct feasible solutions. Starting at time $t = 0$ a truck is sequentially filled with orders until the end of the planning horizon T is reached, or no more order assignment is feasible. At this point another vehicle is brought into use, t is set to $t = 0$ and the order assignment is continued. This procedure is repeated until all orders are assigned.

For the selection of orders that have not yet been assigned to trucks, two aspects are taken into account: how promising the choice of that order is in general, and how good the choice of that order was in previous iterations of the algorithm. The first information is the visibility, the second is stored in the pheromone information.

The proposed ant system can be described by the algorithm given in Table 2.

Table 2: The ACO procedure

```
procedure ACO (η; Γ; Λ) {
  Initialization of the ACO;
  set a number of ants on each depot;
  for Ant := 1 to Γ {
    while not all orders are assigned {
      initialize a new truck;
      t = 0;
      select a home base for the truck;
      while ∃ η_ij(t) > 0      ∀i, j ∈ J {
        select an order using formula (3);
        update t;
      }
    }
    evaluate the objective function;
  }
  for λ := 1 to Λ {
    improve the solution using the post optimization procedure
    evaluate the objective function;
  }
  update local pheromone information;
  return solution_vectors;
}
```

In the initialization phase, Γ ants are generated and each depot is assigned the same number of ants. Then the two basic phases - construction of tours and trail update - are executed for a given number of iterations. To improve the solution quality a post optimization procedure will be applied, which seeks to improve a solution by finding the optimal depot for each truck given the orders assigned.

3.1.1 Visibility

Let J denote the set of orders and D denote the set of depots. The visibility information is stored in a matrix η, each element in the matrix is denoted by $\eta_{ij}(t)$, where $\eta_{ij}(t)$ is positive, if and only if the assignment of order j after order i is feasible. An assignment of order j is feasible, if the order can be scheduled on the current vehicle without violation of its time window. Hence, it is clear that η depends on the time. Note that in each iteration only the row associated with the order assigned in the previous iteration has to be evaluated. The actual value of the visibility of order j depends on the priority rule incorporated in the algorithm. Based on this information we can define the set $\Omega_i(t) = \{j \in J : j$ is an order feasible to assign$\}$.

It is obvious that the choice of the priority rule substantially influences the solution quality. In our problem at hand, we want to minimize total costs, that means a minimization of both empty vehicle movements, as well as number of trucks required. Therefore, the master population of the Cooperative ACO uses a priority rule, which leads to good solutions with respect to total costs. This priority rule takes into consideration the minimization of the empty vehicle movements only insufficiently. Thus, we introduce a slave population, which uses a priority rule suitable to minimize this goal. The relevant information discovered by the ants of the slave population is inserted in the global pheromone information and can be used by the ants of the master population.

3.1.2 Visibility for the master population

The priority rule for the master population is:

$$\eta_{ij}^{master}(t) = \begin{cases} e^{-4\cdot(EDD_j+2\cdot EPST_j(i,t))} & \text{if } j \in \Omega_i(t) \\ 0 & \text{otherwise} \end{cases} \quad \forall j \in J, \tag{1}$$

This priority rule aims to maximize truck utilization by avoiding waiting times. It takes into account the due dates (EDD) and the release dates ($EPST$) of the orders. While the EDD measure exactly represents the due dates, the $EPST$ measure also takes into account waiting times and connecting empty vehicle movements. A more detailed description of the priority rule can be found in [5].

3.1.3 Visibility for the slave population

The priority rule for the slave population is:

$$\eta_{ij}^{slave}(t) = \begin{cases} e^{-16\cdot DIST(i,j)} & \text{if } j \in \Omega_i(t) \\ 0 & \text{otherwise} \end{cases} \quad \forall j \in J. \tag{2}$$

This priority rule is solely based on the distance traveled to get from the delivery location of the last customer assigned (i) to the pickup location of customer j, $DIST(i,j)$. It is obvious, that this priority rule is well suited for the minimization of empty vehicle movements.

3.1.4 Decision rule

Given the visibility and pheromone information, a feasible order j is selected to be visited immediately after order or depot i according to a random-proportional rule that can be stated as follows:

$$\mathcal{P}_{ij}^{\xi}(t) = \begin{cases} \frac{[\tau_{ij}^{\xi}]^{\alpha}[\eta_{ij}^{\xi}(t)]^{\beta}}{\sum_{h \in \Omega_i(t)} [\tau_{ih}^{\xi}]^{\alpha}[\eta_{ih}^{\xi}(t)]^{\beta}} & \text{if } j \in \Omega_i(t) \\ \\ 0 & \text{otherwise,} \end{cases} \quad \forall j \in J, \xi \in \{master, slave\}. \quad (3)$$

This probability distribution is biased by the parameters α and β that determine the relative influence of the trails and the visibility, respectively. τ_{ij}, represents the current pheromone information, i.e. the value τ_{ij} represents the pheromone information of assigning order j immediately after order i.

3.2 Pheromone information

3.2.1 Pheromone Update

After the two ant populations have constructed a feasible solution, the global pheromone trails are updated. We use a pheromone update procedure, where only a number of the best ants, ranked according to solution quality, contribute to the pheromone trails. Such a procedure was proposed in [1] and [2]. The update rule is as follows:

$$\tau_{ij}^{\xi} = \rho \cdot \tau_{ij}^{\xi} + \sum_{\lambda=1}^{\Lambda^{\xi}} \Delta\tau_{ij}^{\xi,\lambda} \text{ , where } \xi \in \{master, slave\} \quad \forall i, j \in J, \quad (4)$$

where ρ is the trail persistence (with $0 \leq \rho \leq 1$). Only the Λ^{master} best ants of the master population and the Λ^{slave} best ants of the slave population update the pheromone information. If an order j was performed immediately after an order

i in the solution of the λ-th best ant of Λ^{master} or of Λ^{slave} the pheromone trail is increased by a quantity $\Delta\tau_{ij}^{\lambda}$. This update quantity can be represented as

$$\Delta\tau_{ij}^{\xi,\lambda} = \begin{cases} 1 - \frac{\lambda-1}{\Lambda^{\xi}} & \text{if } 1 \leq \lambda \leq \Lambda^{\xi}, \text{where } \xi \in \{master, slave\} \\ \\ 0 & \text{otherwise} \end{cases} \qquad \forall i, j \in J. \quad (5)$$

3.2.2 Pheromone Injection

After both ACO procedures are terminated the pheromone injection is performed in order to communicate good solutions from the slave to the master population. Formally this can be written as

$$\tau_{ij}^{master} = \tau_{ij}^{master} + \sum_{\lambda=1}^{\Lambda^{slave}} \Delta\tau_{ij}^{slave,\lambda}. \qquad (6)$$

4 Numerical analysis

In this section we will present the results of our numerical analysis. This analysis is based on a set of testproblems which was generated given the following parameter settings:

- 8 distribution centers,
- 512 orders and
- a planning horizon of 8 periods.

Given these settings we generated 8 problems which differ with respect to the average time window lengths associated with the orders. These average time window lengths were varied between 1 and 8, 1 meaning that every order has to be delivered within one period, while 8 means that no time window restrictions have to be respected.

The objective function to minimize total costs is given by

$$TC = 20 \cdot FS + 1 \cdot MC,$$

where TC denotes the total costs, FS is the fleet size and MC denotes the vehicle movement costs. An interpretation of the weights used in this objective function can be found in [5].

The parameter setting chosen for the Ant System are presented in Table 3.

Table 3: Parameter settings for the ACO algorithm

Parameter	value
α	1
β	1
ρ	0.5
τ_0	0.1
Λ^{master}	8
Λ^{slave}	1
Γ^{master}	128
Γ^{slave}	32
$maxIterations$	30

Let us now turn to the analysis of our cooperative ant colonies. Note, that the results presented in this section are based on averages over five runs for each problem. In order to show the effects of information sharing we compare two different cases.

- Case 1: 1 master population of 160 ants, no slave population

- Case 2: Cooperative ant colonies: 1 master population with 128 ants, 1 slave population with 32 ants, slave population reports good solutions to master population

Case 1 represents a basic ACO algorithm where one population of ants searches the solution space. Each ant in the colony utilizes the same heuristic information, which in our case is the one presented for the master population in the last section. Case 2 represents our new approach which is based on a cooperative system of two ant colonies as presented in the last section. In Table 4 these two cases are compared with respect to empty vehicle movements and fleet sizes. It can be clearly seen, that empty vehicle movements are always less in Case 2. The average improvement in empty vehicle movements is 3.26 %. Furthermore, we see that apart from one problem (the problem with an average time window length of 4 periods) this improvement of the vehicle movements is achieved without detrimental effects on the fleet size. On the contrary, for two problems, those with average time windows of 1 and 6 periods respectively, our new approach even improves the necessary fleet size.

Note, that while the improvements seem to be rather small, the presented results are based on first simulations. Thus, we strongly expect that parameter fine tuning or slight modifications could possibly lead to an increase in these improvements.

5 Conclusions and future research

In this paper we have proposed an ACO approach, where two ant colonies cooperatively solve a multi-objective transportation problem. The objective function was to minimize total costs, consisting of fixed costs for the utilization of the fleet and variable costs for the transportation movements. In our algorithm a large popula-

Table 4: Solution comparison between several advanced Ant System algorithms

Length of Time Windows	Case 1 empty vehicle movements	Case 1 fleet size	Case 2 empty vehicle movements	Case 2 fleet size
1	43.648	28	41.712	27.6
2	31.956	25	31.582	25
3	33.829	26	32.608	26
4	32.496	24.4	29.509	24.8
5	24.313	23	24.108	23
6	25.010	22.2	24.688	22
7	20.563	22	19.956	22
8	18.345	21	17.870	21

tion of ants solves the problem with respect to the fleet size costs, as these costs are the major component in the objective function. This large population is supported by a small population, which aims to minimize total vehicle movement costs and communicates good solutions to the larger master population.

Our results can be viewed as a proof of concept for the proposed method. We showed, that the communication of the two ant colonies improved the solution quality. While these improvements are not very large, they highlight the potential of communication between colonies with different problem solving approaches.

Future research will deal with improvements of this concept, as well as other communication mechanisms which enhance the performance of agent based optimization algorithms.

6 Acknowledgments

The authors are grateful for financial support from the Austrian Science Foundation (FWF) under grant SFB #010 'Adaptive Information Systems and Modelling in Economics and Management Science' and from the Oesterreichische Nationalbank (OENB) under grant #8630.

References

[1] Bullnheimer, B., Hartl, R. F. and Strauss, Ch.: An improved ant system algorithm for the vehicle routing problem. Annals of Operations Research **89** (1999) 319–328

[2] Bullnheimer, B., Hartl, R. F. and Strauss, Ch.: A new rank based version of the Ant System. Central European Journal of Operations Research **7**(1) (1999) 25–38

[3] Colorni, A., Dorigo, M. and Maniezzo, V.: Distributed Optimization by Ant Colonies. In: Varela, F. and Bourgine, P. (Eds.): Proc. Europ. Conf. Artificial Life. Elsevier, Amsterdam (1991)

[4] Costa, D. and Hertz, A.: Ants can colour graphs. Journal of the Operational Research Society **48**(3) (1997) 295–305

[5] Doerner, K. F., Gronalt, M., Hartl, R.F., and Reimann, M.: Optimizing Pickup and Delivery Operations in a Hub Network with Ant Systems. POM Working Paper 07/2000

[6] Dorigo, M.: Optimization, Learning and Natural Algorithms. Doctoral Dissertation. Politecnico di Milano, Italy (1992)

[7] Dorigo, M., Maniezzo, V. and Colorni, A.: Ant System: Optimization by a Colony of Cooperating Agents. IEEE Transactions on Systems, Man and Cybernetics **26**(1) (1996) 29–41

[8] Dorigo, M. and Gambardella, L. M.: Ant Colony System: A cooperative learning approach to the Travelling Salesman Problem. IEEE Transactions on Evolutionary Computation **1**(1) (1997) 53–66

[9] Dorigo, M., Di Caro, G. and Gambardella, L. M.: Ant Algorithms for Discrete Optimization. Artificial Life **5**(2) (1999) 137-172

[10] Gutjahr, W. J.: A graph-based Ant System and its convergence. Future Generation Computing Systems. **16** (2000) 873–888

[11] Irnich, St.: A Multi-Depot Pickup and Delivery Problem with a Single Hub and Heterogeneous Vehicles. European Journal of Operational Research **122**(2) (2000) 310-328

[12] Stützle, T. and Dorigo, M.: ACO Algorithms for the Quadratic Assignment Problem. In: Corne, D., Dorigo, M. and Glover, F. (Eds.): New Ideas in Optimization. Mc Graw-Hill, London (1999)