# A hybrid ACO algorithm for the Full Truckload Transportation Problem

**ARTICLE** · JANUARY 2001

Source: OAI



**3 AUTHORS:**

Karl F. Doerner
University of Vienna


Richard F. Hartl
University of Vienna


Marc Reimann
Karl-Franzens-Universität Graz

# A hybrid ACO algorithm for the Full Truckload Transportation Problem

K. Doerner R.F. Hartl M. Reimann
*Department of Management Science
University of Vienna, Austria

May 10, 2001

**Abstract**

In this paper we propose a hybrid ACO approach to solve a full truckload transportation problem. Hybridization is achieved through the use of a problem specific heuristic. This heuristic is utilized both, to initialize the pheromone information and to construct solutions in the ACO procedure. The main idea is to use information about the required fleetsize, by initializing the system with a number of vehicles rather than opening vehicles one at a time as needed. Our results show the advantages of this new approach over more traditional, i.e. sequential, approaches.

## 1 Introduction

Ant Colony Optimization (ACO), a meta-heuristic which is based on the Ant System introduced in the early nineties of the last century (c.f. Colorni et al. (1991), Dorigo (1992), Dorigo et al. (1996)) has been used to solve a variety of NP hard combinatorial optimization problems, e.g. the vehicle routing problem (Bullnheimer et al. (1999a, 1999b), Gambardella et al. (1999)), the travelling salesman problem (Dorigo and Gambardella (1997)), the graph coloring problem (Costa and Hertz (1997)) and the quadratic assignment problem (Stützle and Dorigo (1999)). Dorigo et al. (1999) provide a thorough overview over problems solved with ACO. Gutjahr (2000) presented a convergence proof for a generalized ant system.

Inspired by the behavior of real ants searching for food, its main idea is to utilize both local information (visibility) as well as information about good solutions obtained in the past (pheromone), when constructing new solutions. In particular this memory is exploited in two ways. First, intensification is achieved by a strong bias towards the best choice in each decision process, based on both pheromone and visibility. Second, diversification is driven by making frequently used paths less desirable to choose.

ACO is based on the problem representation as a graph. The problem is to find the shortest path through the graph, where either all or a pre-specified set of nodes have to be visited. While for some problems, the graphical representation is straightforward (e.g. TSP), for other problems this is not the case. However, the solution quality crucially depends on a proper representation of the problem.

In this paper we propose a new hybrid approach to solve a full truckload transportation problem with time windows. This problem, which was brought to our attention by one of our industry partners, is related to the Vehicle Routing Problem (VRP) and more specifically the Vehicle Routing Problem with Time Windows (VRPTW). The VRP has been solved successfully by Bullnheimer et al. (1999a, 1999b), Gambardella et al. (1999) proposed a multi-colony ant system for the VRPTW. All these approaches are based on a problem representation, where the vehicle depot is duplicated and the problem is to find a least cost TSP route with side constraints (capacity and time) through all orders and the depot nodes. Thus, at a time only one vehicle is considered. A new one is opened, if no more order (customer) can feasibly be assigned to the current vehicle.

* Corresponding author: Marc Reimann, E-mail: marc.reimann@univie.ac.at

Based on these approaches, we initially represented our problem in a similar way and sequentially loaded trucks (c.f. Doerner et al. (2000b)). The problem with these approaches is, that problem specific knowledge is ignored. In particular, one assumes that all orders can be satisfied with one truck. Especially, if fleet sizes are to be minimized, this may lead to poor solutions.

The hybrid approach proposed in this paper, is novel by considering a given fleet size and assigning orders to these vehicles simultaneously. Our objective is to minimize total costs, which consist of costs for both, the utilized vehicle fleet and the total distance traveled. We use a weighted objective function, the main goal is to minimize the fleet size necessary. To achieve this, we initialize the algorithm with a fleet size equal to a lower bound.

Our algorithm consists of a state of the art ACO approach, hybridization is achieved through the use of a heuristic algorithm. First, we use this algorithm to initialize the pheromone matrix. Second, in each decision step we deterministically choose the vehicle to be assigned next, using this heuristic rule. Only the choice of the appropriate order is done by the ACO procedure.

The remainder of this paper is organized in the following way. In section 2 we describe our problem in detail. Section 3 deals with our solution approaches, in particular with our new hybrid ACO approach. The settings and results of our numerical analysis are presented and discussed in section 4. We conclude with some final remarks.

## 2 Description of the problem

As stated in the last section, the problem treated in this paper was brought to our attention by one of our industry partners. The problem, which logistics service providers are typically faced with, is to satisfy a set of transportation requests between a number of distribution centers (DCs). Each order is characterized by its size, it fills a truck completely, and its time windows for pickup and delivery. Given the order size, it is obvious that order consolidation is not an option. Thus, each order is transported directly from its source to its destination. The available fleet is distributed over the distribution centers. Each vehicles utilization is constrained by a maximum tour length restriction, which comes from the necessity, that truck drivers need to return to their home base regularly. The planning horizon is much longer than this maximum possible tour length, such that each vehicle can be used repeatedly within the planning horizon.

The problem described above occurs, as logistics service providers satisfy their customers from a number of distribution centers. Each customer requires a shipment of small size (in relation to the vehicle capacity). If pickup and delivery location are far away from each other, direct transportation is not economical. Thus, customer orders are shipped via one or more hubs. At these hubs or distribution centers order consolidation takes place, such that between the DCs only consolidated transportation orders are shipped. Such a situation is depicted in Figure 1. The shipment of a customer order from location $i$ to location $j$ is performed via the DCs $A$ and $B$. Between the distribution centers $A$ and $B$ a consolidated transportation order consisting of a large number of customer orders is transported.

These consolidated transportation orders, depicted by bold lines, are subject of our optimization as discussed in the first paragraph of this section. A solution approach for the problem 'outside' the hubs, namely between the actual customer sites and the hub, was recently proposed by Irnich (2000).

Our objective is to minimize total costs which consist of fixed and variable vehicle utilization costs. Fixed costs occur as a vehicle is brought into use, independently of the workload assigned to the vehicle. Variable costs correspond to the total distance traveled by the vehicle. We use a weighted function of the two cost terms, where a larger weight is assigned to the fixed costs. Thus, our main goal is to minimize fleetsize, second priority is assigned to the minimization of the vehicle movements. Note in this context, that minimization of the total vehicle movements corresponds to the minimization of the empty vehicle movements as loaded vehicle movements are predetermined by the order structure. This is due to the fact, that because of the order size, all orders are transported directly from their source to their destination. In an earlier paper we proposed a savings based heuristic for the problem to minimize empty vehicle movements (c.f. Gronalt et al. 2000).

Before we continue with the presentation of our algorithmic approach, let us quickly summarize the characteristics of the problem at hand.

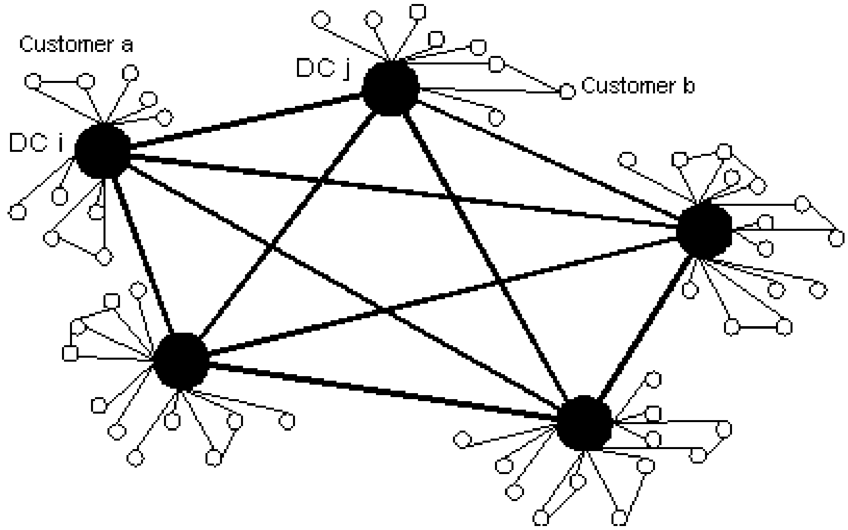


Figure 1: Structure of the distribution network

- All orders are known in advance.
- Each order fills a truck completely.
- Each order requires the shipment between two DCs.
- Time windows at both, pickup and delivery location have to be respected strictly.
- A set of homogeneous vehicles is distributed over the DCs.
- Each vehicle is constrained by a maximum tour length restriction, after which the truck has to return to its home base.
- The planning horizon exceeds this maximum tour length, each vehicle can be used repeatedly.

The problem belongs to the class of pickup and delivery problems. A survey about this problem class can be found in Savelsbergh and Sol (1995). Desrosiers et al. (1988) dealt with a related problem. They proposed an exact algorithm for problem instances without time window restrictions, and were able to solve problems with up to three depots.

# 3 Solution procedures

## 3.1 The constructive heuristic

In this section we present the heuristic solution procedure, which is the core of our new ACO approach. The basic idea is that, in contrast to existing approaches which fill only one vehicle at a time, a certain number of vehicles is considered throughout the algorithm. These vehicles are simultaneously filled with orders. The structure of this heuristic is presented in Table 1.

Here, $J$ denotes the set of orders. The parameters $\tau$, $q_0$ and $\Xi$ are inputs, $\Xi$ denotes the fleet size. The other two parameters are used in the ACO described in section 3.3.

Given the initial fleetsize, orders are assigned to these vehicles until no more assignment is feasible. If all orders are assigned the procedure terminates. In the other case, vehicles are opened as necessary - one at a time - and the assignment is continued until all orders are assigned.

The actual assignment is a two-step process. First, a vehicle is deterministically chosen by considering all feasible order-vehicle assignments and taking the best one according to an attractiveness measure. This measure is described in detail in section 3.3.1. Second, given the choice of the vehicle an order is chosen and a final assignment is done. When all orders are assigned the objective function associated with the solution is evaluated and the solution is returned.

Table 1: The Constructive Heuristic procedure

```
ψ:=procedure ConstructiveHeuristic (τ, q_0, Ξ) {
    while not all orders are assigned {
        compute η_ij^ξ for each truck ξ        ∀i, j ∈ J;
        if ∃ η_ij^ξ > 0 {
        select ξ^best : η^{ξ^best} := maximum(η_ij^ξ);
        }
        else {
            open a further truck Ξ = Ξ + 1;
            ξ^best := Ξ;
            compute η_ij^{ξ^best} for truck ξ^best        ∀i, j ∈ J;
        }
        /* next order j will be assigned to truck ξ^best */
        if (q_0 = 1) {
            select j* : η_{ij*}^{ξ^best} := maximum(η_ij^{ξ^best});
        }
        else {
            select an order using formula (2);
        }
    }
    evaluate the objective function S_ψ;
    return the solution ψ;
}
```

In the next section we will propose our new hybrid ACO approach based on this constructive heuristic.

## 3.2 The new hybrid ACO approach

The hybrid approach we propose aims to exploit the strengths of both the heuristic and the ACO. On the one hand the heuristic exploits the knowledge about the necessary fleetsize, on the other hand the ACO provides the memory to learn from the past.

The hybrid ACO approach can be described by the algorithm given in Table 2.

Table 2: The HybridACO procedure

```
procedure HybridACO {
    SetParameters;
    InitializeSystem;
    generate a number of trucks Ξ according to a lower bound;
    set a number of trucks on each depot;
    /* execute heuristic without pheromone */
    ψ=ConstructiveHeuristic (0, 1, Ξ);
    Initialize pheromone information τ of the ACO
    using solution ψ of the constructive heuristic;
    /* execute ACO */
    ACO (τ; Ξ);
}
```

The procedure `HybridACO` first sets the number of vehicles to the lower bound obtained by some relaxation of the problem. This fleet is distributed over the depots in the next step. Then, the procedure `ConstructiveHeuristic` is called with the parameter $q_0$ set to $q_0 = 1$. Thus, in this first call of the heuristic no random choice of orders occurs. Rather the best order is chosen in each assignment step, leading to a completely deterministic behavior of the heuristic. The heuristic returns the solution, which is then used by the procedure `HybridACO` to reinitialize the pheromone information. Finally, the procedure `ACO` is called, which again utilizes the constructive heuristic to build solutions.

### 3.3 The ACO Algorithm

The proposed ant system can be described by the algorithm given in Table 3.

Table 3: The ACO procedure

```
procedure ACO (τ, Ξ) {
   Initialization of the ACO;
   for i := 1 to max_iterations {
      update Ξ;
      for Ant := 1 to Γ {
         ψ=ConstructiveHeuristic (τ, q_0, Ξ);
         LocalEvaporation(τ, ψ);
      }
      GlobalEvaporation (τ, ψ^{global_best});
   }
}
```

In the initialization phase, $\Gamma$ ants are generated. Then the update of the initial fleetsize and the two basic phases - construction of tours and trail update - are executed for a given number of $max\_iterations$. The initial fleetsize is updated in each iteration. It is set to a value equal to the fleetsize found in the global best solution decremented by one. This approach was inspired by a similar approach proposed in Gambardella et al. (2000). In the construction phase the heuristic proposed in section 3.1 is used to build solutions. Here hybridization is achieved through the following procedure. While the first step in the assignment process of the heuristic - namely the choice of the vehicle - is still done completely deterministic, the second step - namely the choice of an order, given the chosen vehicle - is performed by taking into account both the attractiveness measure and the pheromone information and the decision is based on a pseudo-random-proportional-rule. Therefore, the pheromone information $\tau$ and the parameter setting $q_0$ are also required in the heuristic procedure.

After an ant has constructed its solution, local pheromone evaporation is applied to diversify the search. At the end of each iteration the global pheromone update is performed using the global best solution.

Let us now turn to a more detailed discussion of the visibility and the pheromone information, the decision rule and the pheromone update.

#### 3.3.1 Visibility and Pheromone information

The local attractiveness of assigning an order to a vehicle is called visibility. It depends on the home depot of the vehicle, the order previously assigned to the vehicle and the time at which service of this previously assigned order is finished. The visibility is stored in a three dimensional structure $\eta$, where each layer corresponds to a vehicle. Let $N$ denote the set of depots. Each layer is a matrix with $|J|$ + $|N|$ rows and $|J|$ columns, where the first $|J|$ rows and columns correspond to orders, while the last $|N|$ rows are associated with the depots. Each element in the structure is denoted by $\eta_{ij}^{\xi}$. For $i \leq |J|$, $\eta_{ij}^{\xi}$ denotes the attractiveness of assigning order $j$ to vehicle $\xi$, given that order $i$ was the order previously assigned to the vehicle. For $|J| < i \leq |J| + |N|$, $\eta_{ij}^{\xi}$ denotes the attractiveness of assigning order $j$ as the first order to vehicle $\xi$, which starts from depot $i$. An element $\eta_{ij}^{\xi}$ is positive if and only if order $j$ can be feasibly assigned to vehicle $\xi$, i. e. if it can be scheduled on vehicle $\xi$ without violation of its time window. Thus, we can define the set $\Omega_i^{\xi} = \{j \in J : j$ is an order feasible to assign to vehicle $\xi\}$.

The actual value of the visibility of order $j$ depends on the attractiveness measure incorporated in the algorithm.

It is obvious that the choice of this measure substantially influences the solution quality. For our approach we chose the following measure [1]

[1]The measure given in (1) was chosen after extensive numerical tests.

$$\eta_{ij}^{\xi} = \begin{cases} e^{-32\cdot(WT(j,\xi)+EM_j(i,\xi))} & \text{if } j \in \Omega_i^{\xi} \\ 0 & \text{otherwise} \end{cases} \quad \forall j \in J, \tag{1}$$

where $WT$ denotes a possible waiting time which may occur if the vehicle visits order $j$'s pickup location before the time window for pickup starts, and $EM$ denotes the empty vehicle movements required to reach the pickup location of order $j$.

The waiting times have only an indirect influence on the solution quality as they affect vehicle utilization, which in turn influences the required fleetsize. The empty vehicle movements have both a direct and an indirect influence on the solution quality. First, each movement incurs costs. Second, if the movements exceed a certain limit an additional vehicle has to be utilized. However, there is also a tradeoff between waiting times and empty vehicle movements, as some additional vehicle movements might be avoided by waiting at a location.

The pheromone information is stored in a matrix $\tau$ with $|J|$ + $|N|$ rows and $|J|$ columns. The value $\tau_{ij}$ represents the current pheromone information. If $i \leq |J|$, the value $\tau_{ij}$ represents the pheromone information of assigning order $j$ immediately after order $i$ on the same vehicle. Note, that these values do not depend on the vehicle. While a third dimension were possible to store more accurate information in memory, we believe that the effects on solution quality should be rather small. If $|J| < i \leq |J| + |N|$, the value $\tau_{ij}$ represents the pheromone information of assigning order $j$ as the first order to a vehicle with home depot $i$.

### 3.3.2 Decision rule

Given the visibility and pheromone information, a feasible order $j$ is selected to be visited immediately after order $i$ by truck $\xi$ according to a pseudo-random-proportional rule.

This rule can be stated as follows:

$$\mathcal{P}_{ij}(t) = \begin{cases} max\{[\tau_{ij}]^{\alpha} \cdot [\eta_{ij}^{\xi^{best}}]^{\beta}\} & \text{if } q \leq q_0 \\ \frac{[\tau_{ij}]^{\alpha}[\eta_{ij}^{\xi^{best}}]^{\beta}}{\sum_{h\in\Omega_i^{\xi}}[\tau_{ih}]^{\alpha}[\eta_{ih}^{\xi^{best}}]^{\beta}} & otherwise, \end{cases} \quad \forall j \in J. \tag{2}$$

where $q$ is a random number uniformly distributed in $[0..1)$, $q_0$ is a parameter ($0 \leq q_0 \leq 1$) to be set by the user. This probability distribution is biased by the parameters $\alpha$ and $\beta$ that determine the relative influence of the trails and the visibility, respectively.

### 3.3.3 Pheromone update

- Global Pheromone update

  After each ant of the population has constructed a feasible solution, the pheromone information is updated. We use a pheromone update procedure where only the best solution found so far is used for global updating. The update rule is as follows:

$$\tau_{ij} = (1-\rho)\cdot\tau_{ij} + \rho\cdot\Delta\tau_{ij}, \quad \forall i,j \in J, \tag{3}$$

  where $\rho$ is the evaporation rate (with $0 \leq \rho \leq 1$). If an order $j$ was performed immediately after an order $i$ in the solution of the global-best ant the pheromone information is increased by a quantity $\Delta\tau_{ij}$. This update quantity can be represented as

$$\Delta\tau_{ij} = \begin{cases} S_{\psi^{global\_best}}^{-1} & \text{if } (i,j) \in \psi^{global\_best} \\ 0 & \text{otherwise} \end{cases} \quad \forall i,j \in J. \tag{4}$$

- Local Pheromone update

  After an artificial ant has constructed a feasible solution local updating is performed as follows. When an ant selects an order $j$ after order $i$ the amount of pheromone on the element $\tau_{ij}$ of the pheromone matrix is decreased according to the rule:

$$\tau_{ij} = (1-\rho)\cdot\tau_{ij} + \rho\cdot\tau_0, \quad (5)$$

where $\tau_0$ is the initial value of trails. Local updating enhances diversity in solutions, as ants prefer combinations of orders chosen less frequently in the past.

### 3.4 A standard ACO approach

In the standard ACO approach the function call of the `ConstructiveHeuristic` is replaced by the function call of the `SequentialHeuristic`.

The visibility of the standard ACO approach is implemented as follows. 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.

The sequential heuristic is presented in Table 4.

Table 4: The Sequential Heuristic Procedure

```
procedure SequentialHeuristic(τ) {
  while not all orders are assigned {
    initialize a new truck;
    t = 0;
    select a home base for the truck;
    while ∃ η_ij > 0    ∀i, j ∈ J {
      select an order using formula (2);
    }
    update t;
  }
}
```

## 4 Numerical analysis

### 4.1 Experimental framework

The datasets we used for our numerical analysis were generated to test the performance of our algorithms under different levels of time window strictness.

The basic parameters are:

- 8 distribution centers
- 512 transportation orders
- 8 periods planning horizon
- 2 periods maximum tour length

Given these data, we generated 15 different test instances, which can be grouped into 5 groups. The groups differ with respect to the order structure, i.e. the pickup and delivery locations of the orders. Within each group the order structure is fixed, the problems differ with respect to average time window length. Time windows range between 1 and 8 periods and can take integer values only. To analyze different settings we generated one problem with tight time windows (average time window length is one period), one problem with medium length time windows (average time window length is 4 periods) and one problem without any time windows (average time window length of 8 periods) using binomial distributions. Note, that an average time window length of 1 period corresponds to a situation, where all orders have a time window of 1 period. The other polar case, namely an average time window length of 8 periods, denotes a case where time windows are not restrictive[2].

The objective function considered for our simulations was

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

where $TC$ denotes the total costs, $FS$ denotes the utilized fleet size and $VMC$ denotes the vehicle movement costs. An interpretation of the weights in this objective function can be found in Doerner et al. (2000a).

## 4.2 ACO parameters

The parameter settings for the ACO algorithm are shown in Table 5.

| Parameter | value |
|---|---|
| $\alpha$ | 1 |
| $\beta$ | 1 |
| $\rho$ | 0.1 |
| $\tau_0$ | 0.0001 |
| $q_0$ | 0.9 |
| $\Gamma$ | 16 |
| $maxIterations$ | 100 |

Table 5: Parameter settings for the ACO algorithm

These settings were chosen in accordance with known, good parameter settings from literature (see e.g. Dorigo and Gambardella (1997)).

## 4.3 Results of our simulations

The test instances described above were used to test our hybrid ACO algorithm. We will refer to this approach as 'hybrid ACO'. To evaluate the performance of our novel approach, we compared the results achieved with this approach with results obtained through the use of two alternative methods.

These are

- the heuristic algorithm presented in section 3.1, which we will refer to as 'heuristic'
- a 'sequential' implementation of the ACO algorithm, referred to as 'sequential' ACO.

For the stochastic methods, we generated five runs for each problem instance and the results presented in this section are based on average values over these five runs.

From Table 6 we clearly see that our hybrid ACO approach outperforms both the constructive heuristic and the 'sequential' ACO algorithm. Only for one problem, namely the problem A1, the 'sequential' ACO algorithm finds the best solution. For most of the other problems our new approach is significantly superior to the other methods. What is striking, is that the constructive heuristic outperforms the 'sequential' ACO in 9 instances. If we consider that this algorithm takes approximately 20 seconds to solve the problems, while both ACO procedures run for approximately 90 minutes, the result becomes even more interesting. It shows the strength of the constructive heuristic. Apart from that it shows that problem specific knowledge is necessary if good solutions are to be obtained. By comparing the results for our new hybrid approach with those of the constructive heuristic we see that particularly for the problems with tight time windows, the hybrid ACO approach significantly improves the solutions obtained with the constructive heuristic. In this context, it has to be said, that the problems with tight time windows are those with practical relevance. The results for the problems with wide time windows are presented to show the influence of time window strictness on solution quality.

[2]We only name a single time window for each customer here. Above we spoke of time windows for both pickup and delivery. However, given our problem structure, in particular the assumption that all orders are transported directly from their source to their destination, pickup and delivery time windows can be expressed by a single time window, which specifies the earliest possible pickup time and the latest possible delivery time.

Table 6: Total costs for all instances found by the tested methods

| Instance | heuristic | hybrid ACO | sequential ACO |
|---|---|---|---|
| A1 | 840.2524832 | 781.3356571 | 775.750036 |
| A4 | 609.0964803 | 605.8935363 | 636.3981477 |
| A8 | 587.4652182 | 562.855475 | 608.317732 |
| B1 | 775.2781158 | 723.2951097 | 754.351844 |
| B4 | 602.2157469 | 583.4092565 | 610.9477438 |
| B8 | 584.8718468 | 559.5220581 | 585.7994376 |
| C1 | 775.6004188 | 746.4850722 | 819.783245 |
| C4 | 608.049053 | 582.4896785 | 613.7714801 |
| C8 | 563.7018486 | 558.5965237 | 587.2632497 |
| D1 | 792.3954041 | 735.7684308 | 750.3710276 |
| D4 | 607.344148 | 585.6908365 | 607.9275714 |
| D8 | 586.4834351 | 562.4843707 | 584.8995431 |
| E1 | 789.87508 | 747.2668191 | 766.3796329 |
| E4 | 630.4793283 | 591.0106107 | 625.5415401 |
| E8 | 587.9232173 | 583.5069035 | 611.555753 |

Let us now look more closely at the results, by splitting them into empty vehicle movements and utilized fleetsizes. Table 7 contains this information together with the compulsory loaded vehicle movements and the lower bounds on both empty vehicle movements and fleetsizes. The lower bound for the empty vehicle movements is obtained by relaxing the time constraints and solving a network flow problem. A more detailed description of this lower bound can be found in Gronalt et al. (2000). Given this lower bound on the empty vehicle movements we obtained the lower bound for the fleetsize by evaluating the number of trucks necessary to satisfy the total vehicle movements. In Table 7 the columns associated with the solution procedures contain the empty vehicle movements. The numbers in brackets represent the fleet sizes.

From Table 7 we can see, that our new approach outperforms the other two methods with respect to both empty vehicle movements and fleetsizes. Only in one instance, the 'sequential' ACO algorithm finds the lowest fleetsize, namely for problem instance A1. In another instance the 'sequential' ACO finds less empty vehicle movements than our new approach, however at the cost of an additional vehicle (problem instance B4). Generally the results show, that time window strictness has an important influence on the solution quality. By comparing the problems with average time window lengths of 1 and 8 periods respectively, it can be seen that on average 8 additional vehicles are caused only by the time constraints.

Table 7: Total costs for all instances found by the tested methods

| Instance | Loaded movements | Lower Bound | heuristic | hybrid ACO | sequential ACO |
|---|---|---|---|---|---|
| A1 | 153.22 | 5.983(20) | 50.530(32) | 44.116(29.2) | 47.032(28.6) |
| A4 | 153.22 | 5.983(20) | 23.178(22) | 12.674(22) | 15.876(23) |
| A8 | 153.22 | 5.983(20) | 19.098(21) | 9.635(20) | 14.245(21.8) |
| B1 | 149.33 | 6.877(19) | 45.022(29) | 37.965(26.8) | 45.948(28) |
| B4 | 149.33 | 6.877(19) | 21.618(22) | 14.079(21) | 12.886(22) |
| B8 | 149.33 | 6.877(19) | 16.469(21) | 10.192(20) | 15.542(21) |
| C1 | 150.58 | 4.516(19) | 45.203(29) | 35.905(28) | 45.020(31.2) |
| C4 | 150.58 | 4.516(19) | 23.191(22) | 11.909(21) | 17.469(22) |
| C8 | 150.58 | 4.516(19) | 16.683(20) | 8.017(20) | 13.122(21) |
| D1 | 148.30 | 9.520(19) | 46.0710(30) | 39.468(27.4) | 44.095(27.8) |
| D4 | 148.30 | 9.520(19) | 19.628(22) | 17.391(21) | 19.044(22) |
| D8 | 148.30 | 9.520(19) | 16.599(21) | 14.184(20) | 18.183(21) |
| E1 | 146.14 | 13.017(19) | 48.239(30) | 41.127(28) | 43.735(28.6) |
| E4 | 146.14 | 13.017(19) | 27.402(23) | 20.871(21.2) | 24.339(22.6) |
| E8 | 146.14 | 13.017(19) | 25.416(21) | 17.367(21) | 21.783(22) |

Let us now only consider the problems with an average time window length of 8 periods. By looking at these results, we see that our algorithm finds excellent solutions with respect to fleetsizes, as in the case of an average time window length of 8 periods, the solutions of our new approach are at most two vehicles above the lower bound on the fleetsize. Note however, that the lower bound on the fleetsize depends on the lower bound for the empty vehicle movements as outlined above. Given the movements actually performed in the solution of our hybrid ACO we found the smallest possible fleetsizes for three instances (A8, B8, D8), while for the remaining two instances the results exceeded the smallest possible fleetsize by one vehicle. Finally, we have to point out that the empty vehicle movements are not independent from fleetsizes. On the contrary, it can generally be said, that the minimization of the empty vehicle movements does not lead to solutions which utilize the smallest possible fleetsize. Rather there exists a tradeoff between those two goals, as allowing some additional empty vehicle movements might help to decrease the required fleetsize.

Together these findings suggest, that our algorithm finds good solutions and should be competitive if compared with other approaches. Finally our results show, that the use of sophisticated problem specific knowledge enhances the performance of ACO significantly and should not be ignored if available. While this may lead to a more complex problem representation, as was shown in this paper, the superior solution quality justifies such an approach.

## 5 Conclusion

In this paper we solved a full truckload transportation problem by means of a new hybrid ACO approach, which exploits knowledge about the problem structure, which was ignored in previous approaches for similar problems. In particular it takes into account a information about the required fleetsize.

We showed the advantages of this approach over traditional ACO approaches. Furthermore, we showed that the approach yields good solutions as compared to a lower bound, more specifically the results showed the impact of time window strictness on the solution quality.

Future research will deal with the application of this method to the VRPTW to evaluate the power of this approach more generally. Apart from that, we will exploit the idea presented in this paper to solve dynamic real-world problems, which our industrial partners are faced with.

## 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

Bullnheimer, B., R. F. Hartl, Ch. Strauss. 1999a. An improved ant system algorithm for the vehicle routing problem. *Annals of Operations Research* **89**, 319–328.

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

Colorni, A., M. Dorigo, V. Maniezzo. 1991. Distributed Optimization by Ant Colonies. F. Varela, P. Bourgine, eds. *Proc. Europ. Conf. Artificial Life*, Elsevier, Amsterdam.

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

Desrosiers, J., G. Laporte, M. Sauve, F. Soumis, S. Taillefer. 1988. Vehicle routing with full loads. *Computers and Operations Research* **15**(3), 219–226.

Doerner, K. F., M. Gronalt, R. F. Hartl, M. Reimann. 2000a. Time Constrained Full Truckload Transportation: Optimizing Fleet Size and Vehicle Movements. POM Working Paper 07/2000, Department of Production and Operations Management, University of Vienna, Vienna, Austria.

Doerner, K. F., R. F. Hartl, M. Reimann. 2000b. Ant Colony Optimization applied to the Pickup and Delivery Problem. POM Working Paper 08/2000, Department of Production and Operations Management, University of Vienna, Vienna, Austria.

Dorigo, M. 1992. Optimization, Learning and Natural Algorithms. Doctoral Dissertation, Politecnico di Milano, Italy.

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

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

Dorigo, M., G. Di Caro. 1999. The Ant Colony Optimization Meta-Heuristic. D. Corne, M. Dorigo, F. Glover, eds. *New Ideas in Optimization*, Mc Graw-Hill, London.

Gambardella, L. M., E. Taillard, G. Agazzi. 1999. MACS-VRPTW: A Multiple Ant Colony System for Vehicle Routing Problems with Time Windows. D. Corne, M. Dorigo, F. Glover, eds. *New Ideas in Optimization*, Mc Graw-Hill, London.

Gronalt, M., R. F. Hartl, M. Reimann. 2000. Time Constrained Pickup and Delivery of Full Truckloads. POM Working Paper 02/2000, Department of Production and Operations Management, University of Vienna, Vienna, Austria.

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

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

Savelsbergh, M. W. P., M. Sol. 1995. The general pickup and delivery problem. *Transportation Science* **29**(1), 17–29.

Stützle, T., M. Dorigo. 1999. ACO Algorithms for the Quadratic Assignment Problem. D. Corne, M. Dorigo, F. Glover, eds. *New Ideas in Optimization*, Mc Graw-Hill, London.