Time Constrained Pickup and Delivery of Full Truckloads

Manfred Gronalt[1], Richard F. Hartl[2], Marc Reimann [3]

Department of Production and Operations Management,

Institute for Management Science, University of Vienna,

Bruennerstrasse 72, 1210 Vienna, Austria

Abstract

In this paper we deal with the pickup and delivery of full truckloads (the same analysis applies for the routing of containers, pallets or freight cars) under time window constraints. Our objective function is to minimize empty vehicle movements, as these use resources without directly adding value to the products transported. We first give an exact formulation of the problem. After that we present a relaxed problem formulation based on network flows, which can be used to calculate a lower bound to the solution value. Furthermore, we propose three different heuristics for the problem. Our results show that these heuristics find very good solutions quickly. We also perform a sensitivity analysis by continuously enlarging the planning horizon in order to demonstrate the effects of using advanced information systems for transportation order planning.

[1]e-mail: manfred.gronalt@univie.ac.at

[2]e-mail: richard.hartl@univie.ac.at

[3]Corresponding author: Marc Reimann, 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

1 Introduction and problem description

Due to an ever-growing demand for goods which leads to a rapid increase in transportation volume the efficient use of resources, such as trucks and the road network, becomes more and more important. Efficient vehicle routing can lead to a reduction in the number of trucks needed to provide the service demanded as well as to a better utilization of the street network by reducing vehicle movements.

Generally speaking, one can distinguish between two different problem classes. The first one is concerned with dynamic carrier allocation (e.g. train carriers, pallets) to different locations over time. In this problem only carrier flows are considered. Thus, carriers are not identified individually but are aggregated to groups. The second class of problems deals with the pickup and delivery of orders by a fleet of vehicles, which are clearly identified. This is necessary when tracking and tracing options in telematics systems are used. In such a case firms need to know the location of transportation orders at any times.

Especially important for both classes of problems is the fact that a large fraction of vehicle movements is caused by empty trucks. Dejax et al. [3] found that empty vehicle movements constitute about 40% of total movements.

Considering these findings and forecasts predicting that traffic volume will continue to increase faster than the street network in the next decades, the importance of developing methods to reduce fleet sizes and empty truck movements becomes obvious. The fact that reductions are possible is proved in Du and Hall [5]. In their paper it is shown that efficient algorithms can reduce fleet size significantly.

In this paper we deal with a problem, which belongs to the second class described above. Customers place orders with a logistics service provider, requiring shipments between two locations. The logistics service provider serves these orders from a number of distribution centers. Thus, shipments occur between the pickup location

of an order and the closest distribution center, between distribution centers and between a distribution center and the delivery location of an order.

While shipments between distribution centers and pickup or delivery locations are generally less-than-truckload shipments, movements between different distribution centers are concerned with full truckloads. This is due to consolidation of orders requiring transportation between the same two distribution centers.

The focus of the research presented in this paper is on the problem of delivering full truckloads between different distribution centers, where each distribution center requires pickups as well as drop-offs of goods. Our primary goal is to minimize empty vehicle movements.

The paper is organized as follows. In the next section we present our model and give an exact formulation using an IP approach. Section 4 deals with an LP formulation that can be used to generate lower bounds. In section 5 we propose three heuristics to solve the problem. An illustrative example to show the differences between these three heuristics is presented in section 6. After giving our results in section 7 we conclude with some final remarks and an outlook on opportunities for future research.

2 Related works

Crainic and Laporte [2] and Ball et al. [7] provide general references for a wide variety of fleet management and vehicle routing problems. More specifically, the problem considered in this paper is a variant of the Pickup and Delivery Problem(PDP), for which Savelsbergh and Sol [8] discuss different solution methods, most of which are concerned with less-than-truckload shipments.

For the case of full truckloads an exact approach is discussed in Desrosiers et al. [4]. These authors formulate a model based on an extended graph, where each

order and each vehicle is represented by a node and the movements between orders are modeled as arcs. Furthermore, every location is considered as both depot and demand point. The objective function is to minimize empty vehicle movements. A branch and bound approach based on an algorithm developed for the asymmetrical traveling salesman problem is used to solve the problem without time windows. Results are reported for problems with up to 104 orders and 2 depots. The authors remark that their algorithm is very sensitive with respect to the number of depots as well as the time constraints.

Dumas, Desrosiers and Soumis [6] present an exact formulation and solution method for the less-than-truckload PDP with time windows (PDPTW). They use a set partitioning approach combined with a column generation procedure to find the set of minimum cost routes. A notable feature is the fact that each truck serves only one tour, although another tour might be scheduled within the given time-horizon. In their paper optimal solutions were only found for problem instances with less than 4 depots.

The main difference between these approaches and our problem lies in the fact, that we deal with a multi-period case. Thus, within the planning-horizon each vehicle can be assigned more than one tour. Furthermore, we have to respect time-window constraints, leading to precedence restrictions, which do not occur in the model presented by Desrosiers et al. [4].

Such time windows are considered in Dumas, Desrosiers and Soumis [6]. However, the primary goal of their work is to minimize total routing costs, rather than empty vehicle movement costs. Below we will show that orders are transported directly from their sources to their destinations in our problem, which is not necessarily true for the less-than-truckload case. In the latter case empty vehicle movements can be reduced by performing partly loaded trips.

Due to the differences mentioned, we could not apply these approaches directly to our problem. Thus, although both papers present comprehensive exact solution approaches, we chose to develop a new heuristic algorithm, which is able to find near-optimal solutions for real-life problem instances. This approach is also motivated by the fact that both papers report solutions only for very small problem instances.

3 Model formulation

A fleet of vehicles, based at different distribution centers, has to satisfy a number of orders. Each of these orders needs to be shipped between a pair of distribution centers. All orders are consolidated to full truckloads and are subject to time window constraints at pickup as well as delivery. The underlying assumptions of our model are:

1. All orders are known in advance, the basic problem is static. Thus, planning can be based on full order information. However, this assumption is omitted in one part of our analysis below, in order to show the effects of using advanced information systems for transportation order planning (Furthermore, when applying the model and the solution heuristics, also dynamic problems can be tackled. On the one hand, it is reasonable to use such a planning tool on a rolling basis. On the other hand, it is also possible to quickly evaluate the marginal cost of a new order by calculating the total cost associated both with and without the order and taking the difference.).

2. All orders are consolidated to full truckloads. Thus, we do not deal with cross-docking options. There is at most one order on a truck at any given time, which is transported directly from its pickup location to its delivery location.

3. Time windows for each order have to be respected strictly.

4. A tour must not exceed a given time-span. This assumption models legal restrictions on the maximum driving time for truck drivers. It could also be viewed as a restriction, which ensures that maintenance intervals for vehicles are respected.

5. Each truck is assigned to a specific depot, where it has to return to after each tour. The motivation of this assumption is that truck drivers can't stay away from home too long, but have to come home on a regular basis.

An interesting fact is, that omitting assumptions 3 and 4 leads to a much simpler problem, which can be modeled through network flows. We use this approach to generate lower bounds for our problem. This is explained in more detail in section 4.

The following tour structures are possible under these assumptions. Single order tours occur if a truck returns unloaded to its origin after delivering an order. In this case 50% of the tour length consist of empty vehicle movements. If possible, tours with multiple orders will be performed. Connecting orders can be done in one of three ways. First, if a particular truck is moved from distribution center A to distribution center B it can pick up another order at location B and deliver it to any other location. Second, it can be immediately forwarded to any other location to pick up an order there. The third option for a truck is to wait at B until a new transportation order is due at that location. Thus, in this third option waiting times occur in addition to transportation times. Furthermore, these options lead to three different types of vehicle movements, namely compulsory trips of full trucks, connecting trips of empty trucks and empty trips to close tours.

The exact formulation for our problem is given now.

- Parameters:

  A ... set of orders

  K ... set of tours

  F_i ... time needed to perform order i, transportation and loading/ unloading time $\forall i \in A$

  E_{ij} ... time needed to go from the delivery location of order i to the pickup location of order j $\forall i, j \in A$

  LFT_i ... latest delivery time for order i $\forall i \in A$

  EST_i ... earliest pickup time for order i $\forall i \in A$

  $MaxDuration$ ... maximum duration of a tour

  Note, that satisfying all orders on separate tours is the most expensive planning option, which also utilizes the maximum number of vehicles. Thus, in the worst case the number of tours is equal to the number of orders, such that the set of tours K has the same cardinality as the set of orders A. However, if some feasible solutions are known, e.g. from the application of some heuristics, the set K can be reduced accordingly.

- Decision variables:

  $x_{ik} \in \{0, 1\}$ ...1, if order i is assigned to tour k

  $y_{ij} \in \{0, 1\}$ ...1, if order j is scheduled to be performed immediately after order i, **or** order j is the last and order i the first order on a tour

  $s_{ik} \in \{0, 1\}$ ...1, if order i is performed before all other orders on tour k

  $e_{ik} \in \{0, 1\}$ ...1, if order i is performed after all other orders on tour k

  t_i^s ...effective pickup-time for order i

  t_i^e ...effective delivery-time for order i

  τ_k^s ...starting time for tour k

τ_k^e ...ending-time for tour k

$d_k \in \{0, 1\}$ ...1, if tour k is performed

- Objective function:

$$min \sum_{i \in A} F_i + \sum_{i \in A} \sum_{j \in A} E_{ij} y_{ij} \tag{1}$$

- Constraints:

$$\sum_{k \in K} x_{ik} = 1 \qquad \forall i \in A \tag{2}$$

$$d_k \geq x_{ik} \qquad \forall i \in A, \forall k \in K \tag{3}$$

$$s_{jk} \leq x_{jk} \qquad \forall j \in A, \forall k \in K \tag{4}$$

$$e_{jk} \leq x_{jk} \qquad \forall j \in A, \forall k \in K \tag{5}$$

$$\sum_{j \in A} s_{jk} = d_k \qquad \forall k \in K \tag{6}$$

$$\sum_{j \in A} e_{jk} = d_k \qquad \forall k \in K \tag{7}$$

$$x_{ik} - x_{jk} \leq 1 - y_{ij} \qquad \forall i, j \in A (i \neq j), \forall k \in K \tag{8}$$

$$x_{jk} - x_{ik} \leq 1 - y_{ij} \qquad \forall i, j \in A (i \neq j), \forall k \in K \tag{9}$$

$$s_{ik} + e_{jk} - 1 \leq y_{ji} \qquad \forall i, j \in A, \forall k \in K \tag{10}$$

$$\sum_{i \in A} y_{ij} = 1 \qquad \forall j \in A \tag{11}$$

$$\sum_{j \in A} y_{ij} = 1 \qquad \forall i \in A \tag{12}$$

$$t_i^e \leq LFT_i \qquad \forall i \in A \quad (13)$$

$$t_i^s \geq EST_i \qquad \forall i \in A \quad (14)$$

$$t_i^s + F_i = t_i^e \qquad \forall i \in A \quad (15)$$

$$t_i^e + E_{ij} - M \sum_{k \in K} e_{ik} \leq t_j^s + M(1 - y_{ij}) \qquad \forall i, j \in A (i \neq j) \quad (16)$$

$$\tau_k^s \leq t_i^s + M(1 - s_{ik}) \qquad \forall i \in A, \forall k \in K \quad (17)$$

$$\tau_k^e \geq t_j^e + E_{ji} - M(2 - s_{ik} - e_{jk}) \qquad \forall i, j \in A, \forall k \in K \quad (18)$$

$$\tau_k^e - \tau_k^s \leq MaxDuration \qquad \forall k \in K \quad (19)$$

M is as usual a 'sufficiently large' positive number (e.g. the length of the planning horizon in our case).

The objective function (1) minimizes the sum of loaded and empty vehicle movements. Note, that the sum of loaded movements is a constant factor in this case, as all orders are transported directly from their pickup to their delivery locations and these trips are compulsory. Thus, we actually minimize empty vehicle movements.

Constraints (2) assign each order to exactly one tour. Constraints (3) require that a tour is performed if at least one order is assigned to it, while constraints (4), (5), (6) and (7) ensure that first and last orders of each tour are chosen among the orders assigned to that tour and that each tour which is performed has exactly one first and one last order. Constraints (8) and (9) deal with the fact that two orders can only be connected if they are assigned to the same tour. Constraints (10) ensure that a vehicle movement from the delivery location of a tour's last order to the pickup location of the tour's first order is performed. Constraints (11) ensure for every order, that its pickup location is approached from the delivery

location of exactly one other order. Constraints (12) ensure for every order that a vehicle movement from its delivery location to the pickup location of exactly one other order occurs. Together constraints (10), (11) and (12) imply that each tour is closed, and that each order can be preceded and followed by at most one other order. Constraints (13) and (14) ensure that the latest possible delivery-time and the earliest possible pickup-time are respected. Constraints (15), (16), (17) and (18) constitute the temporal reality: Constraints (15) require for each order, that the difference between pickup and delivery time of the order equals the time needed to perform that order. Constraints (16) require that the difference between the effective delivery time of one order and the effective pickup time of a second order is at least as large as the time needed to go from the first order's delivery location to the second order's pickup location, if the two orders are scheduled to be performed immediately after each other. Constraints (17) ensure that each tours starting time is not later than the pickup time of the first order assigned to the tour. Constraints (18) require that the ending time of a tour is not earlier than the delivery time of the last order assigned to the tour plus the time needed to go back to the tour's starting location. Finally, constraints (19) ensure that the maximum possible duration of a tour is respected.

Generally, MIP's (like the one we presented here) belong to the class of NP-hard problems and efficient exact algorithms to solve these problems for realistic problem sizes do not exist. Thus such problems can only be solved by heuristics which can lead to suboptimal solutions of a priori unknown quality. In such a case it is useful to find good lower bounds, in order to get an estimate for the quality of the solution found by the heuristic procedure. As stated above, the loaded movements of the vehicles are fixed, such that only empty vehicle movements can be optimized. Thus, in the next section we will present a lower bound for empty vehicle movements.

4 A lower bound for empty vehicle movements

To get an estimation of the minimum share of empty vehicle movements in total vehicle movements, we propose the following lower bound. Remember that every order requires the shipment of a full truckload between two locations within a given time-window. Considering also, that every truck has to return to its starting location after finishing its tour, this means that for every location, the number of incoming flows must be equal to the number of outgoing flows. The times at which these flows occur are interdependent, as a truck can only leave a location after it has arrived at it. However, if we relax the time constraints (13) - (19) in the IP formulated above, the objective function (1) and constraints (2) - (12) can be reformulated as a network flow LP.

Based on the following parameters

N ... number of distribution centers

c_{ij} ... time needed for a movement between locations i and j $\forall i, j \in N$

D_{ij} ... number of necessary loaded movements between locations i and j $\forall i, j \in N$

and the decision variables

S_{ij} ... Total number of movements (loaded and empty) from i to j $\forall i, j \in N$

our LP formulation can be written as

$$min \sum_{i \in N} \sum_{j \in N} c_{ij} S_{ij} \tag{20}$$

such that:

$$S_{ij} \geq D_{ij} \qquad \forall i, j \in N \tag{21}$$

$$\sum_{i \in N} S_{ij} = \sum_{k \in N} S_{jk} \qquad \forall j \in N \tag{22}$$

The objective function (20) minimizes total vehicle movements, both loaded and empty. Constraints (21) ensure that all loaded vehicle movements are performed, while constraints (22) require that at each distribution center the number of incoming vehicle movements is equal to the number of outgoing vehicle movements.

Note that the solution of this relaxed problem is also the optimal solution of the actual problem if it satisfies the relaxed time constraints. We will use this lower bound to analyse the performance of the three heuristics presented in the next section. Throughout our analysis we will assume that the costs and times needed for vehicle movements are directly proportional, and use these terms interchangeably.

5 Heuristic solution procedures

In this section we will present three heuristic algorithms for the problem discussed above. The first algorithm is based on the well known Savings Algorithm proposed by Clark and Wright [1]. The second and third algorithm introduce opportunity costs into the solution procedure of our first algorithm.

5.1 A Modified-Savings Algorithm

The Savings-Algorithm was proposed as a solution method for the Vehicle Routing Problem(VRP). The VRP is the problem of finding a set of minimal cost routes for a fleet of vehicles, which have to visit every member of a set of customers with known demand exactly once. The total demand of all customers exceeds the vehicle capacity, thus not all customers can be served on one giant tour. The basic VRP deals with identical vehicles, all based at the same location, where they have to return to after completion of their tours. Constraints limiting the maximum length of a tour may exist.

For the VRP, the Savings Algorithm starts with the initial allocation of each customer to a separate tour. Then for each pair of customers the cost-savings of joining those customers on one tour is calculated. Based on the values of these savings the customers are sequentially joined into tours starting with the customer combination yielding the largest cost-savings and proceeding until no more savings can be achieved.

In this problem all customers either demand deliveries (of goods) or pickups (e.g. disposal of waste). Thus, if customers demand deliveries the single pickup location of orders is the depot, where the vehicles are based. In analogy to that the single delivery location of orders is the depot, if customers demand pickups.

Clearly, the Savings algorithm in its original formulation cannot be used for our problem since we have multiple depot pickups and deliveries here. However, we can construct an algorithm which is based on some kind of priority numbers which can also be interpreted as savings. As we will show below, in our case it is not sufficient to calculate all savings the savings only once at the initialization of the algorithm. Rather, we have to recalculate some of the savings after each move of connecting orders or sub-tours:

1. Savings calculation

2. Savings update

In our algorithm savings are calculated between orders, rather than between locations. This is necessary, as with each order two locations are associated. Thus, if we think of two orders i and j, the first of which needs to be picked up at location A and delivered to location B, while the second needs to be shipped between

locations C and D respectively, the cost-savings of joining these orders on one tour subsequently can be calculated as

$$s(i,j) = cost(B,A) + cost(D,C) - cost(B,C) - cost(D,A) \tag{23}$$

where $cost(B,A)$ denotes the costs of traveling between locations B and A. This move is shown in Figure 1. Figure 1a shows the necessary movements before the orders are joined, while Figure 1b shows the resulting movements after the orders have been joined.

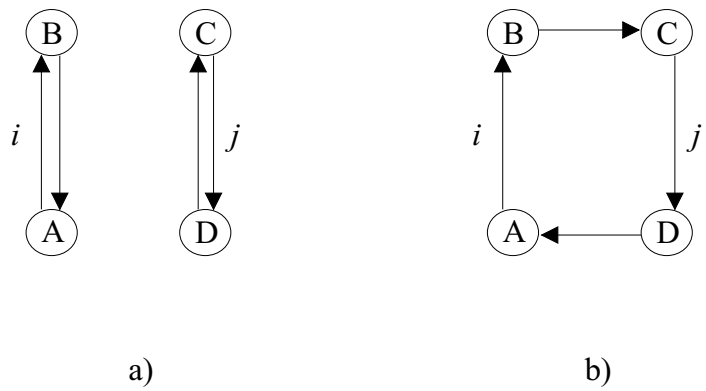


Figure 1: Joining two orders i and j on one tour.

Furthermore, the joining of two orders leads to the necessity of re-calculating all the savings associated with those orders. Consider for example, the savings value between orders j and k, where order k is to be picked up from location A and delivered to location C. In this case the savings value prior to joining orders i and j is

$$s(j,k) = cost(D,C) + cost(C,A) - cost(D,A) \tag{24}$$

Note that in this case the tour is closed with order k, as order k's destination location is the pickup location of order j. However, after orders i and j have been

joined this is no longer the case. Joining order k with orders i and j leads to savings of

$$s(j,k) = cost(D,A) + cost(C,A) - cost(D,A) - cost(C,A) = 0. \tag{25}$$

This is due to the fact that now we are actually calculating the savings between a tour (consisting of orders i and j) and order k. The starting location of the tour is the pickup destination of order i, which is A. The last delivery of the tour is to be made at location D, as this is the delivery location of order j. Thus, we calculate the savings between a tour starting at A and ending at D and order k, which is exactly the savings value given by (25). More generally, after each iteration the savings values associated with the first and the last order of the tour constructed in this iteration have to be updated.

Below, Figure 2a shows the initial situation. Figure 2b depicts the resulting movements, if only orders j and k are joined on one tour. Figure 2c shows the changed situation, if orders i and j have been joined already. In this case joining order k to these two orders yields no further improvements.

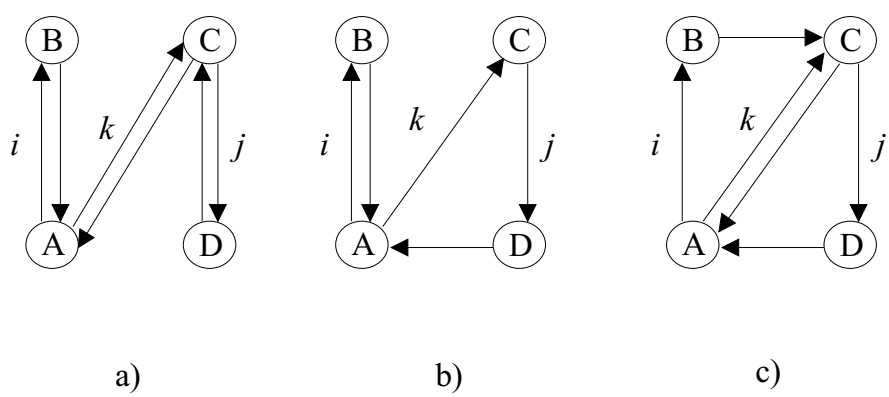


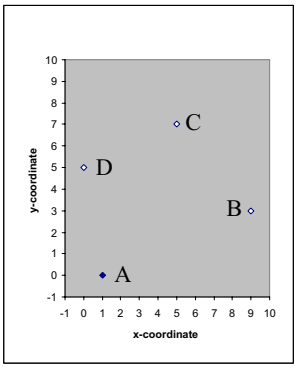
Figure 2: Joining two orders j and k on one tour.

5.2 Opportunity-Savings Algorithms

These algorithms are a modification of the algorithm described above. Savings are not solely chosen according to their absolute value. Instead, for each order we generate the opportunity costs of not choosing the largest savings value associated with this order, but the second largest. Then we choose the largest savings value for the order with the largest opportunity costs. We will point out the differences between the two Opportunity-Savings Algorithms and the Modified-Savings Algorithm in detail, using the illustrative example below.

6 An illustrative example

We will consider an example with four distribution centers, four orders and a planning-horizon of 8 periods. The maximum length of a tour must not exceed 2 periods. For this setting Figure 3a shows the geographical location of the distribution centers, while Figure 3b depicts the distance matrix.



a)

	A	B	C	D
A	0	0,46	0,44	0,28
B	0,46	0	0,31	0,5
C	0,44	0,31	0	0,29
D	0,28	0,5	0,29	0

b)

Figure 3: Geographical location of the distribution centers(a) and distance matrix(b).

The order data are given in the following table.

Order #:	1	2	3	4
Pickup location:	B	B	B	D
Delivery location:	C	D	A	C
EST	2	0	1	3
LFT	3	6	4	8

Table 1: Order data.

For each order, Table 1 shows the pickup and delivery locations, as well as the earliest possible pickup time and the latest possible delivery time. While order 1 has a narrow time-window of only one period, order 2 can be satisfied at any time between the first and the sixth period. It can be seen from the table, that order 4 can't be performed immediately before order 1 on a tour, as the time-window of order 4 begins after the time-window of order 1 ends.

If all these orders are scheduled on separate tours, the total vehicle movement costs equal 3.124.

Savings	1	2	3	4	opportunity costs
1					0,000
2				0,485	0,000
3				0,172	← 0,172
4		0,485			0,000

Table 2: Savings and opportunity costs for the Modified-Savings Algorithm and the Opportunity-Savings Algorithm V1.

The possible cost-savings are shown in Table 2. Due to the structure of the problem (three of the four orders have the same pickup location) only few positive savings values exist. Order 2 can be scheduled either immediately before or imme-

diately after order 4 on the same tour. Additionally order 3 can be scheduled before order 4. In addition to the savings values, Table 2 also shows the opportunity costs for each order of not choosing the best savings value. For order 3 these opportunity costs are exactly equal to the cost-savings associated with scheduling order 3 immediately before order 4. The reason is, that if this combination is not chosen, order 3 cannot be combined with any other order. Orders 2 and 4 can be combined with each other in two ways. Thus, for order 2 opportunity costs are zero, as failing to schedule order 2 immediately after order 4 still leaves the opportunity of scheduling these orders the other way round. The same applies for order 4. Order 1 cannot be scheduled with any other order, hence no opportunity costs exist.

For this example the differences in the choice mechanisms of our three heuristics can be easily explained. Let us first consider the Modified-Savings Algorithm. For this algorithm, combinations are chosen according to their savings value, starting with the combination yielding the largest cost-savings. In Table 2 this value can be found in the dark grey shaded cell. Thus, according to the Modified-Savings Algorithm, order 2 should be scheduled immediately before order 4 on the same tour. If this combination is chosen, no further savings can be achieved, and the final solution is found. Compared with the case, where all the orders are scheduled on separate tours, the cost-savings are 0.485, thus the total vehicle movement costs are 2.629.

Let us now look at the Opportunity-Savings Algorithms. The first variant of the algorithm picks the order with the largest opportunity costs of not choosing the best combination for this order. In our example these largest opportunity costs can be found in the light grey shaded cell. For this order, which in our case is order 3, the combination yielding the largest cost-savings is chosen. In the example, this leads to the combination of orders 3 and 4 on one tour, where order 3 is scheduled

immediately before order 4. This choice, leads to the situation, that no further cost-savings can be realized, hence the total savings in this case are 0.172. In this case the total vehicle movement costs are 2.952. From now on, we will refer to this algorithm as Opportunity-Savings V1.

The second variant of the algorithm calculates the opportunity costs for rows and columns separately. The value associated with each row gives the opportunity costs of not scheduling an order before its best successor. The value associated with each column shows the opportunity costs of not joining an order with its best predecessor. The choice based on this algorithm is shown in Table 3. From now on, we will refer to this algorithm as Opportunity-Savings V2.

Savings	1	2	3	4	opportunity costs
1					0,000
2				0,485	0,485
3				0,172	0,172
4		0,485			0,485
opportunity costs	0,000	0,485	0,000	0,313	

Table 3: Savings matrix for Opportunity-Savings V2.

It can be seen that for our small problem this version of the Opportunity Savings Algorithm finds the same solution as the Modified-Savings Algorithm. The opportunity costs for not choosing to pair order 2 with its best successor are 0.485. The same opportunity costs are associated with not choosing to schedule order 2 immediately after its best predecessor (which is order 4) as well as with not choosing to schedule order 4 immediately before its best successor (which is order 2). All these cells are shaded dark grey. However, if we search the savings matrix sequentially we first find the value associated with order 2 and its best successor. Thus, we choose this value and order 2 is scheduled immediately before order 4 on one tour.

Thus, in our simple example, the Modified-Savings Algorithm and Opportunity-Savings Algorithm V2 found the same result with lower costs than Opportunity-Savings Algorithm V1. To find out how these algorithms perform for more realistic problem sizes we performed a numerical study. In the next section the results of this study are presented.

7 Numerical results

7.1 A comparison of our heuristics

In order to analyse the performance of our heuristics we tested them on a number of randomly generated problems, for which we used the following input parameters:

- 8 distribution centers,
- 64 orders,
- a maximum tour length of 2 periods and
- a planning horizon of 32 periods, the number of periods for which order information is available.

Based on this information x- and y-coordinates from the interval (0,100] were randomly assigned to each distribution center. The geographical setting of our test-problems can be seen in Figure 4.

Thus, all our test-problems have the geographical data in common. We chose this approach for practical reasons. It is obvious that firms can't change the locations of their warehouses in the short run. Thus, given these locations, firms have to satisfy sets of orders which change over time. Under this constellation, of changing

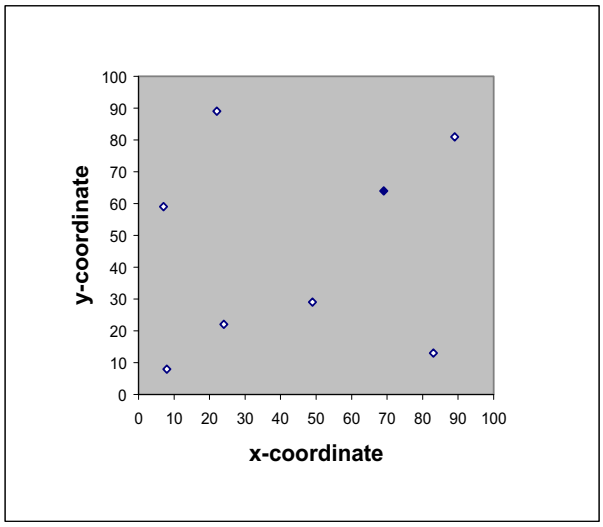


Figure 4: Geographical location of the distribution centers.

orders sets, it is important that a solution method yields robust results which do not heavily depend on the order portfolio. However, our results are robust with respect to different geographical settings as well.

The orders are specified by a pickup and a delivery location, both chosen from the set of distribution centers, as well as an earliest possible pickup time and a latest possible delivery time, both randomly generated within the interval (0,32]. As stated above, 32 is the number of periods for which order information is available and thus planning can be done.

Table 4 shows the results of our LP relaxation (the lower bounds), and the results of our three heuristics for ten representative problem instances.

The first important finding, which can be seen from Table 4 is the fact, that the Modified-Savings Algorithm outperforms each of the two Opportunity-Savings Algorithms in 9 out of 10 instances. Only for problem instance 5 the Opportunity-Savings Algorithm V1 yields slightly better results, while for problem instance 9 the Opportunity-Savings Algorithm V2 finds the best solution. However, columns

Cols.	1	2	3	4	5	6	7	8	9
Prob.#	loaded trips [1]	Lower Bound	% [2]	Savings	% [3]	Opportunity-Savings V1	% [4]	Opportunity-Savings V2	% [5]
1	17,422	20,571	15,31	22,414	8,22	23,337	11,85	23,239	11,48
2	18,391	20,536	10,45	21,802	5,81	21,827	5,91	24,315	15,54
3	17,311	20,413	15,20	21,410	4,66	22,290	8,42	21,751	6,15
4	18,359	24,614	25,41	25,744	4,39	26,147	5,86	26,263	6,28
5	19,749	22,915	13,82	24,676	7,14	24,607	6,88	26,619	13,91
6	18,047	22,189	18,67	23,011	3,57	24,017	7,61	23,508	5,61
7	17,327	20,921	17,18	22,017	4,98	23,200	9,82	22,156	5,57
8	18,627	22,099	15,71	23,020	4,00	23,044	4,10	24,104	8,32
9	17,332	20,038	13,50	22,573	11,23	23,120	13,33	22,125	9,43
10	17,794	19,512	8,80	20,620	5,37	21,575	9,56	21,261	8,23
Avg.			15,40		5,94		8,34		9,05

[1]: Entries in this column show the total costs of loaded vehicle movements.
[2]: These percentage values are calculated as (x2-x1)/x2.
[3]: These percentage values are calculated as (x4-x2)/x4.
[4]: These percentage values are calculated as (x6-x2)/x6.
[5]: These percentage values are calculated as (x8-x2)/x8.
xn denotes the entry in row x and column n.

Table 4: Results of our three heuristics as compared to the lower bound.

5 and 7 show, that the Savings algorithm finds solutions which are on average 5.94% above the lower bounds, while the solution values of the Opportunity-Savings Algorithm V1 lie on average 8.34% above the lower bounds. Overall this algorithm yields slightly better results than the Opportunity-Savings Algorithm V2, which on average finds solutions that lie 9.05% above the lower bound.

Considering the costs of empty vehicle movements it is obvious from the table that on average these costs constitute at least 15.4% of total costs, as this is the average difference in percent between lower bound and costs of loaded movements as found in column 3. Together with the fact, that our Modified-Savings Algorithm finds solutions, which lie on average 6% above the lower bounds, it can be concluded that using advanced solution concepts for transportation order planning leads to empty vehicle movement costs, which amount for approximately 22% of

total vehicle movement costs. If we compare these results with the findings of Dejax et al. [3], where empty vehicle movement costs amounting to 40% of total costs were reported for a large European transportation system, we can conclude that intelligent computational planning efforts can lead to improved and more efficient use of resources.

7.2 Availability of information and planning results

The results given so far were all computed using a planning horizon of 32 periods. The next part of our analysis is concerned with the effects of changing information regarding the order situation. Starting with a scenario, where only the order information for the next day is available, we continuously enlarge the planning horizon, thus providing more information as a basis for planning. The aim is to show the value of information for transportation order planning. While most firms use highly sophisticated systems to generate data, only a small fraction of the information available is actually used for planning purposes.

Table 5 shows the solutions associated with planning horizons of 1, 2, 4, 8 and 16 periods as well as the solutions for 32 periods. A planning horizon of one means that only the next day's order information is available. Thus, planning is done on a daily basis. The other polar case, namely that of 32 period planning means that all the information for the next 32 periods is available and used for planning. For this case, we already presented the results above. Nevertheless, they are listed in Table 5 to show the differences between various levels of information. For the cases of 2, 4, 8 and 16 periods the same intuition applies, as for the two polar cases described above.

The results shown in Table 5 were obtained using the Modified-Savings Algorithm. It is obvious that long-range planning yields much smaller costs than

Cols.	1	2	3	4	5	6	7	8	9	10	11	12	13
Prob.#	loaded trips	S(1) [1]	% [2]	S(1) [1]	% [2]	S(4) [1]	% [2]	S(8) [1]	% [2]	S(16) [1]	% [2]	Savings S(32) [1]	% [2]
1	17,422	30,243	42,39	28,303	38,44	25,756	32,36	24,194	27,99	23,017	24,31	22,414	22,27
2	18,391	32,500	43,41	29,963	38,62	27,648	33,48	25,523	27,94	24,997	26,43	21,802	15,65
3	17,311	31,267	44,63	29,121	40,55	26,314	34,21	25,688	32,61	23,100	25,06	21,410	19,15
4	18,359	32,781	43,99	31,458	41,64	27,000	32,00	26,932	31,83	26,212	29,96	25,744	28,69
5	19,749	34,081	42,05	29,614	33,31	28,522	30,76	25,990	24,01	25,008	21,03	24,676	19,97
6	18,047	30,663	41,14	28,509	36,70	27,054	33,29	25,365	28,85	24,613	26,68	23,011	21,57
7	17,327	30,941	44,00	28,660	39,54	25,686	32,54	24,734	29,95	23,589	26,55	22,017	21,30
8	18,627	33,037	43,62	29,572	37,01	27,403	32,03	25,038	25,61	23,683	21,35	23,020	19,08
9	17,332	30,501	43,18	29,407	41,06	27,711	37,45	26,127	33,66	23,470	26,15	22,573	23,22
10	17,794	29,696	40,08	27,537	35,38	26,206	32,10	24,543	27,50	21,539	17,39	20,620	13,71
Avg.			42,85		38,23		33,02		29,00		24,49		20,46

[1]: Entries in these columns give the total costs associated with the Savings solution for a certain planning time-horizon. The length of this time-horizon is given as the number in brackets.

[2]: Entries in these columns show the share of empty vehicle movement costs from total costs for the associated Savings solution.

Table 5: Results of the Savings algorithm for varying planning horizons.

day-to-day planning. While the average costs for empty vehicle movements are more than 42% of the total costs in the case of day-to-day planning, these costs amount for less than 21% of total costs, if one plan is made for all 32 periods. The decrease in costs can also be seen in Figure 5. From this figure it becomes obvious that increased information about order data leads to significantly better planning results. However, the marginal value of information decreases with increasing planning horizons. The higher the initial level of information is, the less impact has more information on the planning results.

All our simulations were run on a Pentium with 120 MHz. Computational times for the Savings Algorithm were less than ten seconds.

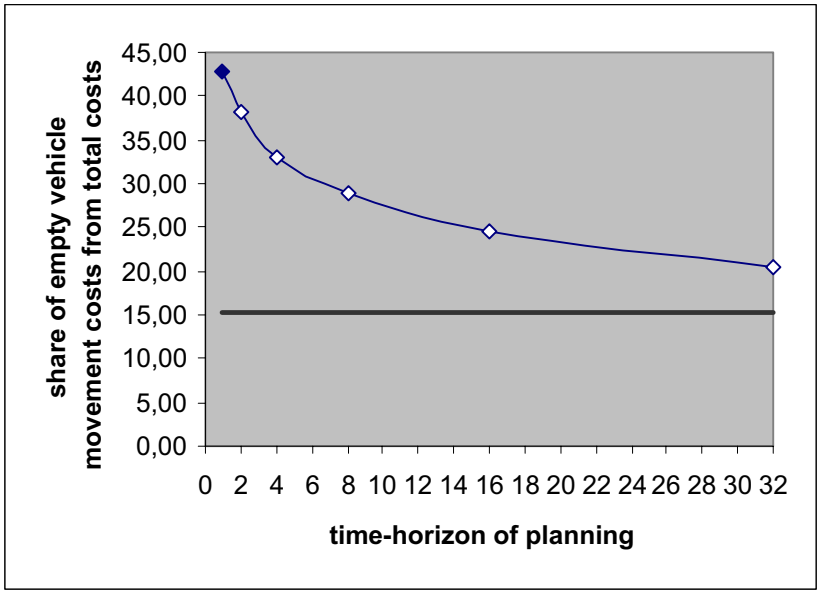


Figure 5: Empty vehicle movement costs for different planning-horizons.

8 Conclusions and future research

In this paper we presented an exact formulation for an NP-hard problem in distribution logistics. We proposed a relaxation of the problem to generate lower bounds, which we used to test three new heuristics. Our results show that these heuristics find very good solutions quickly.

We also showed the value of information, by providing an analysis of the costs associated with different lengths of the planning horizon. Considerable differences between the costs for empty vehicle movements were reported for different levels of information, indicating that increased levels of information lead to better plans and thus to cost reductions.

Future research will be concerned with algorithms which explicitly minimize the fleet size necessary to satisfy all customer's orders, as well as explicit consideration of carrier allocation to certain locations over time.

References

[1] G. Clark, J. W. Wright, Scheduling of vehicles from a central depot to a number of delivery points, Operations Research 11 (1963) 568–581.

[2] T. G. Crainic, G. Laporte, (Eds.), Fleet management and logistics, Kluwer Academic Publishers, Norwell, 1998.

[3] P. J. Dejax, T. Durand, F. Servant, A planning model for loaded or empty container ground transportation, TIMS/ORSA national meeting, Los Angeles, 1986.

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

[5] Y. Du, R. Hall, Fleet sizing and empty equipment redistribution for center-terminal transportation networks, Management Science 43(2) (1997) 145–157.

[6] Y. Dumas, J. Desrosiers, F. Soumis, The pickup and delivery problem with time windows, European Journal of Operational Research, 5(1) (1991) 7–22.

[7] M. O. Ball. et al., (Eds.), Handbooks in Operations Research and Management Science, Volume 8: Network Routing, North-Holland, Amsterdam, 1995.

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