

## Integrated planning of loaded and empty container movements

Peer-reviewed author version

BRAEKERS, Kris; CARIS, An & JANSSENS, Gerrit K. (2013) Integrated planning of loaded and empty container movements. In: OR SPECTRUM, 35 (2), p. 457-478.

DOI: 10.1007/s00291-012-0284-5

Handle: <http://hdl.handle.net/1942/14633>

# Integrated planning of loaded and empty container movements\*

Kris Braekers<sup>a</sup>, An Caris<sup>a,b</sup> and Gerrit K. Janssens<sup>a</sup>

<sup>a</sup>Transportation Research Institute, Hasselt University  
Campus Diepenbeek, Wetenschapspark gebouw 5  
3590 Diepenbeek, Belgium

<sup>b</sup>Research Foundation Flanders (FWO)  
Egmontstraat 5, 1000 Brussels, Belgium  
e-mail: {kris.braekers,gerrit.janssens,an.caris}@uhasselt.be

## Abstract

Efficiently planning drayage operations is an important task for transportation companies since these operations constitute a large part of the cost of an intermodal transport. In this paper a full truckload vehicle routing problem for transporting loaded and empty containers in drayage operations is studied. For empty container transports either the origin or the destination is not predefined. The problem is formulated as an asymmetric multiple vehicle Travelling Salesman Problem with Time Windows (*am-TSPTW*). Two solution approaches are proposed: a sequential and an integrated approach. For both approaches, a single and a two-phase deterministic annealing algorithm are presented. Results show that the proposed algorithms are able to find good quality solutions in a small amount of computation time. The integrated approach clearly outperforms the sequential one and the results confirm the advantage of using a two-phase algorithm for vehicle routing problems with hierarchical objectives. Finally, it is shown that the proposed integrated solution method improves previous results on a similar problem.

Keywords: Vehicle routing, Empty containers, Drayage, Deterministic Annealing

## 1 Introduction

In the hinterland of a major seaport, drayage operations are concerned with the transport of loaded and empty containers between shippers, consignees, inland container terminals and container terminals at the port. Efficiently planning these operations is an important task for shipping lines and transportation companies since they represent a considerable amount of the total cost of an intermodal transport (Smilowitz, 2006). Special attention should be paid to minimizing empty container movements since they are costly non-revenue generating activities.

---

\*This article is published in *OR Spectrum* (2013), 35(2), 457-478 (DOI:10.1007/s00291-012-0284-5). The original publication is available at [www.springerlink.com](http://link.springer.com/article/10.1007/s00291-012-0284-5): <http://link.springer.com/article/10.1007/s00291-012-0284-5>.

Traditionally, a sequential approach is used for the operational planning of loaded and empty container movements. The problem is decomposed into two subproblems, an allocation and a routing problem. First, an empty container allocation model is used to determine the optimal repositioning of empty containers based on the locations of demand and supply in the region. Such an allocation model minimizes the total distance travelled by empty containers, without taking vehicle routing decisions into account. Next, a routing model is used to create efficient vehicle routes performing both loaded and empty container transport requests (Crainic et al, 1993; Huth and Mattfeld, 2009). The objective is to minimize the number of vehicles used, minimize the distance travelled, minimize travelling time or a combination of these. Such vehicle routing models are known to be very complex, especially when time windows are involved (Cordeau et al, 2007). To solve problems of realistic size often meta-heuristics, such as tabu search or simulated annealing, are used.

Palmgren et al (2003) describe a similar decomposition approach for the log truck scheduling problem in the forest industry. The authors solve a transportation problem to find the possible flow between supply and demand points. When creating feasible routes for their problem, the results of the transportation problem are used to limit the number of demand points that can be reached from a certain supply point.

Recently, some efforts are made to integrate the allocation and routing subproblems in drayage operations described above. By using an integrated approach, thus considering empty container allocation and vehicle routing decisions simultaneously, drayage costs may be reduced. Since the origin or destination of empty container transports are not determined in advance, the resulting problem is even more complex. Although several papers have addressed this idea, the advantage of an integrated over a sequential approach for planning drayage operations, has not been quantified.

In this paper both a sequential and integrated approach are presented and compared with each other for a full truckload drayage problem with time windows. For both approaches, the problem is formulated as an asymmetric multiple vehicle Travelling Salesman Problem with Time Windows (*am-TSPTW*) and a single and a two-phase deterministic annealing algorithm are proposed to solve the problem. Results clearly show that the integrated approach outperforms the sequential one. Besides, results in this paper confirm the advantage of using a two-phase approach for vehicle routing problems with the hierarchical objective of first minimizing the number of vehicles used and second total distance travelled. Finally, to assess the quality of the proposed solution method, it is shown that our algorithm improves previous results on a similar problem.

In Section 2, an overview of related literature is given. A detailed problem description, together with an overview of both solution approaches, is presented in Section 3. In Section 4, the proposed deterministic annealing algorithms are described. The experimental design and results are discussed in Section 5. Finally, conclusions are drawn and future research opportunities are identified in Section 6.

## 2 Literature review

In this section, related literature is presented. First, literature on empty container allocation models and full truckload routing is discussed. Next, recent approaches to integrate empty container allocation and vehicle routing decisions are described.

## 2.1 Empty container allocation models

The objective of an empty container allocation model is to determine the best distribution of empty containers, while satisfying both known and forecasted demand (Crainic et al, 1993). Besides empty container demand and supply, repositioning empty containers in order to be able to satisfy empty container demand in future periods may be considered. Container allocation models are described among others in Chang et al (2008); Chu (1995); Crainic et al (1993); Di Francesco et al (2006); Julia et al (2003) and Olivo et al (2005). For a detailed review of these and other empty container allocation models the reader is referred to Braekers et al (2011).

## 2.2 Full truckload routing

Routing problems in drayage operations can be classified as full truckload pickup and delivery problems (Erera and Smilowitz, 2008). The routing problem considered in this paper is a deterministic Full Truckload Pickup and Delivery Problem with Time Windows (FT-PDPTW), which can be reduced to an asymmetric multiple vehicle Travelling Salesman Problem with Time Windows (*am*-TSPTW).

Gronalt et al (2003) develop four savings based heuristics for a FT-PDPTW. Goods are transported between distribution centers or depots. Vehicles are based at different depots and may perform several routes during the planning period. A FT-PDPTW with multiple vehicle depots and additional weight constraints in the context of log truck scheduling is studied by Gronalt and Hirsch (2007). Different variants of the tabu search meta-heuristic are proposed to solve the problem. Imai et al (2007) introduce a full truckload pickup and delivery problem in the context of an intermodal terminal. Caris and Janssens (2009) extend this problem to a FT-PDPTW by including time window constraints at the customer locations. The problem is solved by a local search heuristic. In a subsequent work, a deterministic annealing algorithm is proposed to solve the problem (Caris and Janssens, 2010). The effect of the introduction of an appointment-based access control system at a port on full truckload drayage operations with time windows is studied by Namboothiri and Erera (2008). Mes et al (2007; 2010) propose an agent-based approach for a dynamic version of the FT-PDPTW.

Julia et al (2005) and Wang and Regan (2002) show that a FT-PDPTW may be transformed to an asymmetric multiple vehicle Travelling Salesman Problem with Time Windows (*am*-TSPTW) by collapsing each transport request into a single node. Wang and Regan (2002) use a time window partitioning method to solve the problem. The authors iteratively solve an under- and over-constrained version of the problem. Julia et al (2005) present two exact approaches for solving small instances of the *am*-TSPTW, using dynamic programming and genetic algorithms. An insertion heuristic is proposed to solve large problem instances. Lower bounds on the number of vehicles for the (*a*)*m*-TSPTW are presented by Desrosiers et al (1988) using Lagrangian relaxation, and by Mitrović-Minić and Krishnamurti (2006) using precedence graphs.

The Multiple Depot Vehicle Scheduling Problem with Time Windows (MDVSPTW), which is equivalent to a FT-PDPTW with multiple vehicle depots, is studied by Mingozzi et al (1995); Desaulniers et al (1998) and Hadjar and Soumis (2009). Currently, problems up to 900 tasks and can be solved to optimality. However, since the problem is applied in the context of urban bus scheduling, time windows are assumed to be small (maximum 30 minutes) while in our problem time windows up to four hours are considered.

## 2.3 Integrated approaches

Dejax and Crainic (1987) already suggested that the independent consideration of container allocation and vehicle routing neglects possible synergies arising from an integrated view on these problems. However, Crainic et al (1993) stated that a single mathematical model comprising container allocations and vehicle routing would be computationally intractable. Due to the continuous improvement of Operations Research techniques and computer capabilities, this opinion has changed. Recently, a number of attempts have been made to integrate container (or trailer) allocation and vehicle routing decisions.

Baldacci et al (2006) study the Multiple disposal facilities and multiple inventory locations Rollon-Rolloff Vehicle Routing Problem (M-RRVRP). The problem arises in the sanitation industry where tractors move trailers between customer locations, disposal facilities and inventory locations. Five types of trips are identified. For some trip types, the origin or destination of an empty trailer is not predefined. A set partitioning formulation is used and an exact solution method is proposed. Deidda et al (2008) propose a static, deterministic optimization model simultaneously addressing the allocation of empty containers between shippers, consignees and a port and the design of vehicle routes for empty containers. Loaded container transports are not considered. Vehicles are located at a single depot at the port and have a capacity of two containers. An exact algorithm is proposed. Huth and Mattfeld (2009) compare the results of a sequential and an integrated decision making approach for allocating and routing swap containers in a hub-and-spoke network. Vehicle routing decisions for both loaded and empty containers with a vehicle capacity of two containers are considered. The allocation problem is modelled as a multi-stage transportation problem while the routing problem is modelled as a generalization of the pickup and delivery problem. A Large Neighborhood Search (LNS) is used to solve the routing model in both the sequential and integrated approach. Results show the advantage of an integrated approach for this type of problem.

Some integrated approaches for full truckload (drayage) problems with time windows have been proposed as well. Currie and Salhi (2004) propose a tabu search heuristic for a FT-PDPTW with heterogeneous products and vehicles where the pickup points of goods to be delivered to customers are not predefined. The objective is to minimize total costs, including a fixed cost per vehicle used. Smilowitz (2006) studies the routing and scheduling of loaded and empty trailers or containers in drayage operations. Trailer allocation decisions are made simultaneously with vehicle routing decisions by introducing flexible tasks for empty trailers demanded and supplied (origins resp. destinations are not predefined). Only allocations with a distance smaller than a threshold are considered as possible executions for a flexible task. The objective of the model is to both minimize fleet size and travel time. The model is solved by a branch-and-bound heuristic using column generation. This solution method is improved in a subsequent work (Francis et al, 2007). Recently, dynamic versions of this problem are studied by Escudero et al (2011) and Zhang et al (2011a). Another column generation approach embedded in a branch-and-bound framework for optimizing drayage operations of trailers is proposed by Ileri et al (2006). A heterogeneous fleet of drivers with different start and end locations is assumed. The objective is to minimize costs with company drivers having a different cost structure than third party drivers. When travelling between certain types of tasks, intermediate stops at empty container storage and supply locations are introduced to ensure that vehicles arrive at the starting location of a task appropriately (empty or with an empty trailer). A sim-

ilar problem, in the context of container transportation, is investigated by Zhang et al (2009). Empty container allocations are integrated with routing decisions for vehicles with a single container capacity. A single container terminal and several vehicle depots with an empty container stock are considered. Vehicles do not need to return to their starting depot. The objective is to minimize total travel time. It is shown that the problem can be formulated as a multiple vehicle Travelling Salesman Problem with Time Windows (*m*-TSPTW) and multiple depots. A Reactive Tabu Search (RTS) algorithm is proposed to solve the problem. Zhang et al (2010) extend this problem to a multiple depot, multiple terminal problem and solve it by a time window partitioning method. Finally, Zhang et al (2011b) look at the single depot, single terminal problem where the number of empty containers available at the depot is limited. Again an RTS algorithm is proposed. It seems that solving this problem is much more complex than solving the problems in Zhang et al (2009) and Zhang et al (2010) where there is no limit on the number of empty container available at the depots.

### 3 Problem description

The problem in this paper is to create efficient vehicle routes performing all loaded and empty container transport requests in a region during a single day. It is assumed that a single vehicle depot and one or more container terminals are located in the region. Both the container terminals and the vehicle depot are opened during the whole planning period. Empty containers can be stored at each container terminal and sufficient empty containers are available at each terminal.

A loaded container transport represents a transport from a shipper to a container terminal (outbound loaded container) or from a container terminal to a consignee (inbound loaded container). For each container, the terminal to be used is predefined so that for all loaded container transports the origin and destination are known in advance. Time windows are imposed on these transport requests.

For empty container transports, either the origin or the destination is not defined in advance. A shipper may request an empty container to be delivered before a specific point in time. The origin of this empty container is irrelevant for the shipper and can be chosen by the decision maker. On the other hand, a consignee will have an empty container available after unloading an inbound loaded container. This container becomes available at a certain point in time and should be picked up before the end of the day. The destination of the empty container is determined by the decision maker. Empty containers can thus be transported from consignees to a container terminal, from a container terminal to a shipper or directly from a consignee to a shipper.

A homogeneous fleet of vehicles with a single container capacity is assumed. All vehicles start and end their route at the vehicle depot. When a vehicle arrives early at a location, waiting is allowed at no cost. The service time to load and unload containers is constant and the same for loaded and unloaded containers. A hierarchical objective function is used. The primary objective is to minimize the number of vehicles used while the secondary objective is to minimize total distance travelled.

The problem described above is similar to the ones studied in Zhang et al (2009) and Zhang et al (2010), although some differences exist. The objective is twofold in this paper (first minimize vehicles, then minimize distance) while respectively only total travelling time and total operation time are minimized in Zhang et al (2009) and Zhang et al (2010). Besides, when a vehicle delivers a loaded container to a consignee, it does not have to wait at this location until the container is unloaded and can be picked

up. Instead, in this paper it is assumed that the vehicle may leave for another task and an empty container that becomes available at the consignee's location may be picked up by any vehicle. Finally, in this paper only a single vehicle depot is assumed and the empty container stocks are located at the container terminals rather than at the vehicle depot.

In the following sections, the sequential and integrated solution approaches are described in detail. To make this discussion more clear, a small example is shown in Figure 1 to demonstrate how a solution is found by the two solution approaches. Figure 1(a) shows a small problem situation. The network consists of a vehicle depot, two container terminals with an empty container stock, a single loaded container that has to be transported from one of the terminals to a consignee, a single empty container supply location and a single empty container demand location. No time windows and a single vehicle are considered in this example. Parts (b), (c) and (d) of Figure 1 are discussed throughout Sections 3.1 and 3.2.

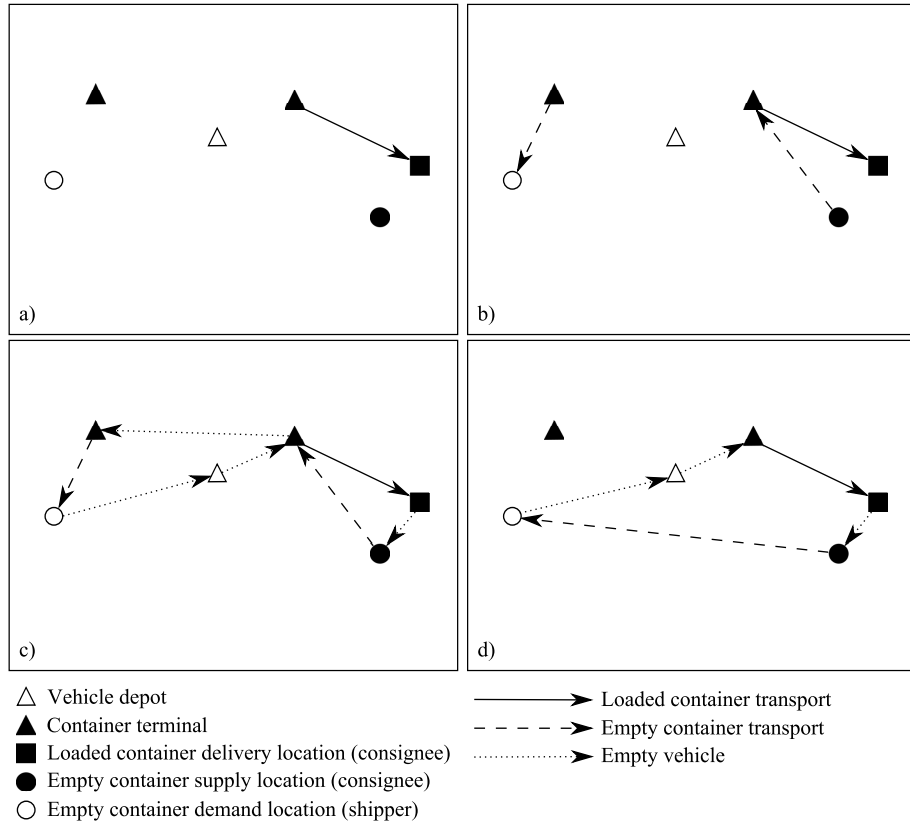


Figure 1: Illustration of both solution approaches

### 3.1 Sequential approach

When solving the problem sequentially, empty container allocations are determined before vehicle routes are created. This may lead to a suboptimal solution but reduces the complexity of the vehicle routing problem. The allocation model is discussed in

Section 3.1.1. This model results in a set of empty container transports that need to be performed. In a second step, the routing problem is solved for loaded and empty container transports together. This problem is described in Section 3.1.2.

### 3.1.1 Empty container allocation problem

Based on known demand and supply locations, the best distribution of empty containers is determined by an allocation model. The objective is to minimize the total distance travelled by empty containers.

The problem is formulated as a transportation problem. The set of origins is composed of the container terminals and the empty containers supplied by consignees. Similarly, the set of destinations is composed of the container terminals and the empty containers demanded by shippers. For each container terminal, supply at the origins and demand at the destinations are set to respectively the number of empty containers demanded and supplied in the region. For each possible allocation, the cost is represented by the distance between the origin and destination. The cost for allocations that are not feasible because of time window violations is set to a very large value. Finally, terminal-terminal allocations have a cost of zero since they do not represent actual movements.

The transportation problem is solved optimally by the well-known Ford-Fulkerson algorithm (Ford and Fulkerson, 1956). The optimal allocations (consignee-shipper, consignee-terminal and terminal-shipper) represent the empty container transports that need to be performed. These transports now have a fixed start and end location. For each transport, the time window for the start of service at the origin is determined based on the time windows of the origin and destination.

Solving the empty container allocation problem for the example in Figure 1 leads to the optimal (least-distance) solution shown in part (b) of this figure. The empty container supplied is transported to the container terminal on the right while the empty container demanded is supplied from the container terminal on the left. Another option is to perform a street turn by transporting the empty container supplied directly to the empty container demand location, but then the distance traveled by the empty container would be larger.

### 3.1.2 Routing problem

Once the empty container transport requests are known, the routing problem becomes a Full Truckload Pickup and Delivery Problem with Time Windows (FT-PDPTW). Since vehicles are assumed to have a single container capacity, the origin and destination of a transport request should be visited immediately after each other by the same vehicle. Therefore, a transport request may be represented by a single node and the problem can be formulated as an asymmetric multiple vehicle Travelling Salesman Problem with Time Windows (*am*-TSPTW) (Jula et al, 2005; Wang and Regan, 2002).

The following notation is used. Let  $G = (N, A)$  be a graph with node set  $N = \{0, 1, \dots, n\}$  and arc set  $A = \{(i, j) | i, j \in N, i \neq j\}$ . The set of nodes consists of a vehicle depot (index 0) and  $n$  tasks or transport requests to be performed. Each task  $i \in N \setminus \{0\}$  has:

- a start location,
- an end location,



- a distance  $d_i$  which is the distance to travel directly between the start and end location,
- a duration  $s_i$  which is the sum of the loading time, the travel time from start to end location and the unloading time,
- a time window  $[a_i, b_i]$  between which the node should be reached by a vehicle in order to be able to perform the task in time.

For the vehicle depot ( $i = 0$ ),  $d_i$  and  $s_i$  are equal to zero and the time window is equal to the total planning period  $[0, b_0]$ . It is assumed that the distance and travel time between two locations are proportional to the Euclidean distance between these locations.

The parameters  $d_{ij}$  and  $t_{ij}$  represent respectively the distance and time to travel from the end location of task  $i$  to the start location of task  $j$ . The maximum number of vehicles available is  $K$  and  $M$  is a very large number (at least  $b_0 + \max_{(i,j) \in A} t_{ij}$ ).

Binary decision variables  $x_{ij}$  are used to determine whether any vehicle  $v \in V$  travels directly from the end location of task  $i$  to the start location of task  $j$ . Continuous variables  $t_i$  represent the point in time at which a vehicle starts task  $i$ . The routing problem is formulated as follows:

$$\text{lex min}(\sum_{i \in N} x_{0i}, \sum_{(i,j) \in A} d_{ij}x_{ij} + \sum_{i \in N} d_i) \quad (1)$$

Subject to

$$\sum_{j \in N} x_{ij} = 1 \quad \forall i \in N, i \neq 0 \quad (2)$$

$$\sum_{j \in N} x_{0j} \leq K \quad (3)$$

$$\sum_{j \in N} x_{ij} = \sum_{j \in N} x_{ji} \quad \forall i \in N \quad (4)$$

$$t_i + s_i + t_{ij} \leq t_j + M(1 - x_{ij}) \quad \forall (i, j) \in A, j \neq 0 \quad (5)$$

$$t_i + s_i + t_{i0} \leq b_0 + M(1 - x_{i0}) \quad \forall i \in N \quad (6)$$

$$a_i \leq t_i \leq b_i \quad \forall i \in N \quad (7)$$

$$t_i \geq 0 \quad \forall i \in N \quad (8)$$

$$x_{ij} \in \{0, 1\} \quad \forall (i, j) \in A \quad (9)$$

As discussed in Section 3, a hierarchical or lexicographic objective function is used (1). The primary objective is to minimize the number of vehicles used ( $\sum_{i \in N} x_{0i}$ ). The secondary objective is to minimize total distance travelled, which consists of the distance travelled from each task to the next ( $\sum_{(i,j) \in A} d_{ij}x_{ij}$ ) and the distance travelled to perform each task itself ( $\sum_{i \in N} d_i$ ). Constraints (2), (3) and (4) are flow constraints. Constraint (5) ensures that a vehicle cannot start a new task before finishing the previous task and travelling to the new one. Constraint (6) ensures that all vehicles return to the vehicle depot before the end of the planning period. Time windows are represented by constraint (7). Finally, constraints (8) and (9) make sure that both types of variables only take on the appropriate values.

Figure 1(b) shows that after solving the empty container allocation problem for the small example, three transport tasks need to be performed: one loaded container transport and two empty container transports. Using the vehicle routing problem described above leads to the optimal solution for the sequential approach which is shown in Figure 1(c).

Table 1: Node parameters

Nodes	Distance $d_i$	Duration $s_i$	Time window $[a_i, b_i]$
$i \in N_{VD}$	0	0	$[0, b_0]$
$i \in N_L$	Distance between start and end location	Loading time + travel time between start and end locations + unloading time	$[a_i, b_i]$
$i \in N_S$	0	Loading time	$[a_i, b_0]$
$i \in N_D$	0	Unloading time	$[0, b_i]$

### 3.2 Integrated approach

When using an integrated approach, empty container allocations are not made beforehand but simultaneously with vehicle routing decisions. This means that the origin (destination) of an empty container demanded by a shipper (supplied by consignee) is not fixed in advance.

The integrated problem can be formulated by creating a node for the vehicle depot, for all loaded container transports and for all feasible empty container allocations. Extra constraints should impose that exactly one allocation node for each consignee and for each shipper is chosen. Such a formulation of the problem can be found in Braekers et al (2010). Solving this problem is however problematic since the number of possible allocations and thus the number of nodes in the network becomes very large for problems of a realistic size. Smilowitz (2006) uses a similar approach by defining feasible allocations as possible executions of a flexible task. To overcome the problem of the exponential growth of the network, a heuristic column generation approach is proposed and the number of feasible allocations is restricted by imposing a maximum distance.

Alternatively, the integrated problem can be formulated like the routing problem for the sequential approach. This is done by introducing an intermediate stop at a container terminal when travelling between certain types of nodes (Ileri et al, 2006; Zhang et al, 2009). The node set  $N$  is composed of the vehicle depot ( $N_{VD}$ , index 0), a set of nodes for the loaded container transport ( $N_L$ ), a set of nodes for the empty container demand locations ( $N_D$ ) and a set of nodes for the empty container supply locations ( $N_S$ ).

Again, to each node is assigned a distance  $d_i$ , a duration  $s_i$  and a time window  $[a_i, b_i]$ . For the vehicle depot and the nodes representing the loaded container transports, these values are identical as those for the sequential approach. For the empty container supply and demand locations  $d_i$  is zero and  $s_i$  is equal to the (un)loading time. Time windows are  $[0, b_i]$  for demand locations with  $b_i$  the latest time the container should be delivered at node  $i$  and  $[a_i, b_0]$  for supply locations with  $a_i$  the time that the container becomes available for pickup node  $i$ . An overview of these values is shown in Table 1. The calculation of the distance  $\hat{d}_{ij}$  of travelling between two nodes  $i$  and  $j$  is shown in Table 2, where  $r$  is the number of container terminals in the region. When an intermediate stop at a container terminal needs to be made, the choice for the terminal is based on a lowest distance basis. Travel times  $\hat{t}_{ij}$  between two nodes are calculated similarly as the distances, but augmented with the container loading and unloading time when making an intermediate stop at a container terminal or travelling directly from an empty container supply to an empty container demand location. The other variables  $(K, M, x_{ij}, t_i)$  and the formulation of the problem are the same as for the

Table 2: Calculation of distance coefficients  $\hat{d}_{ij}$ 

	$j \in N_{VD} \cup N_L$	$j \in N_S$	$j \in N_D$
$i \in N_{VD} \cup N_L$	$d_{ij}$	$d_{ij}$	$\min_{e=1,\dots,r}(d_{ie} + d_{ej})$
$i \in N_S$	$\min_{e=1,\dots,r}(d_{ie} + d_{ej})$	$\min_{e=1,\dots,r}(d_{ie} + d_{ej})$	$d_{ij}$
$i \in N_D$	$d_{ij}$	$d_{ij}$	$\min_{e=1,\dots,r}(d_{ie} + d_{ej})$

sequential approach, except that variables  $d_{ij}$  and  $t_{ij}$  are replaced by respectively  $\hat{d}_{ij}$  and  $\hat{t}_{ij}$ .

Although the formulation is the same, the integrated problem is harder to solve than the routing problem of the sequential approach. The reason is twofold. First, the number of nodes slightly increases. Second, the nodes representing the empty container demand and supply locations have much wider time windows than the nodes representing empty container transport requests in the sequential approach.

When the integrated approach is used to solve the small example in Figure 1(a), the optimal vehicle route is determined without first deciding on the origin and destination of the empty container demanded respectively supplied. Figure 1(d) shows the optimal solution. Clearly the total distance traveled is less than for the solution of the sequential approach (shown in Figure 1(c)). The reason for this lower distance is that another empty container allocation is chosen. In this case, an empty container is transported directly from the empty container supply location to the empty container demand location. Although the distance traveled by empty containers is larger for the integrated solution (since the optimal container allocation is not chosen), total distance traveled by the vehicle is smaller than for the solution of the sequential approach.

## 4 Deterministic annealing algorithms

In the previous section it is shown that for both solution approaches an *am*-TSPTW needs to be solved. Solving such a problem exactly for realistic problem sizes is very hard (Jula et al, 2005). Meta-heuristics can be used to find good and often optimal or near-optimal solutions. In this paper, a deterministic annealing algorithm is proposed.

Deterministic annealing is a variant on the well-known simulated annealing meta-heuristic. With deterministic annealing, also known as threshold accepting, a neighboring solution worsening the objective function is accepted if the worsening is smaller than a certain threshold (Dueck and Scheuer, 1990). This threshold may be adapted during the search. Deterministic annealing has been proven more effective than its stochastic counterpart simulated annealing for several problems. Recently, deterministic annealing has been successfully implemented for a number of vehicle routing problems (Bräysy et al, 2003; 2008; Caris and Janssens, 2010; Nikolakopoulos and Sarimveis, 2007; Tarantilis et al, 2004).

In Section 4.1, the insertion heuristic used to find a feasible starting solution for the algorithm is discussed. The local search operators and the deterministic annealing scheme are respectively described in Sections 4.2 and 4.3. Finally, the implementation of the algorithm on the problem under study is discussed in Section 4.4.

## 4.1 Insertion heuristic

An initial solution for the problem is obtained by a parallel insertion heuristic similar to the one proposed by Julia et al (2005). Starting from a number of vehicles equal to the lower bound, a node is randomly selected and inserted in its best possible position. This operation is repeated until all nodes are inserted. If no insertion is possible, an additional vehicle is added. The procedure is iterated a number of times to obtain a good starting solution.

## 4.2 Local search operators

Several local search operators are used to improve the initial solution. Three operators try to reduce the distance travelled. With the *relocate* operator, a node is removed from its route and reinserted in another route or in another place in its original route. The *2-Opt\** operator (Potvin and Rousseau, 1995) removes an arc from two routes and recombines the resulting parts, that is: the first part of route one with the second part of route two and vice versa. The *exchange* operator swaps a number of nodes between two routes. Several variations of this operator are used, based on the number of nodes that are swapped: *exchange(1,1)*, *exchange(2,1)*, *exchange(2,2)*, *exchange(3,2)* and *exchange(3,3)*. Only one of these exchange operators is used in an iteration of the algorithm. For the *exchange(1,1)* operator the insertion position of a node can also be one place before or after the removal position of the other node. When two nodes from a route are swapped to another route, the reverse insertion of these nodes is considered as well. For each operator, one route is selected randomly and all combinations of nodes or arcs of this route with those of the other routes are considered. A first improvement strategy is used.

Finally, two operators try to reduce the number of vehicles by reinserting all nodes of respectively one or several routes into the other routes. The first operator tries to insert all nodes of a randomly selected route into the other routes. The second operator tries to insert all nodes of the  $p$  shortest routes into all other routes and  $p - 1$  empty routes. The parameter  $p$  is defined as a percentage of the number of routes in the solution.

## 4.3 Deterministic annealing scheme

The proposed deterministic annealing scheme is based on the one in Bräysy et al (2008) and Caris and Janssens (2010). It is shown in Algorithm 1. The current best solution  $S_{best}$  is set to the best solution found by the insertion heuristic and the threshold  $T$  is set to its maximum value  $T_{max}$ . The deterministic annealing algorithm is iterated  $n$  times. At each iteration, all local search operators are used in a random order. New solutions with a lower number of vehicles as the current solution are always accepted. New solutions with the same number of vehicles and total distance lower than the distance of the current solution plus the threshold  $T$  are accepted. If a solution is better than the best solution found so far, this solution is set as best solution. When no new best solution has been found, the threshold  $T$  is reduced by the threshold reduction parameter  $\Delta T$ . Whenever  $T$  becomes negative, it is reset to  $r \times T_{max}$ , with  $r$  a random number between zero and one. In case  $T$  becomes negative and no improvement has been found for  $n_{imp}$  iterations, the search is restarted from the best solution.

---

**Algorithm 1** Implementation of the deterministic annealing algorithm

---

$T = T_{max}$ ,  $i_{last} = 0$  and  $S_{best}$  = best solution of the insertion heuristic

**for**  $i = 1 \rightarrow n$  **do**

**for**  $j = 1 \rightarrow m$  **do**

    Apply local search operator on  $S$  and accept or reject new solution  $S'$

**if**  $S'$  is accepted **then**

$S \leftarrow S'$

**if**  $f(S) < f(S_{best})$  **then**

$S_{best} \leftarrow S$

$i_{last} \leftarrow 0$

**end if**

**end if**

**end for**

**if**  $i_{last} > 0$  **then**

$i_{last} \leftarrow i_{last} + 1$

$T \leftarrow T - \Delta T$

**if**  $T < 0$  **then**

$T \leftarrow r \times T_{max}$

**if**  $i_{last} > n_{imp}$  **then**

$S \leftarrow S_{best}$

$i_{last} \leftarrow 0$

**end if**

**end if**

**end if**

**end for**

---

## 4.4 Implementation

In this section, the implementation of the deterministic annealing algorithm on the problem described in Section 3 is discussed. For both the sequential and integrated approach a single and a two-phase solution algorithm are presented.

### 4.4.1 Single phase algorithms

The implementation of the single phase algorithms is straightforward. An initial solution is found by the insertion heuristic and this solution is improved by the deterministic annealing algorithm for a predefined number of iterations. All local search operators are used during the algorithm which means that the number of vehicles and total distance are minimized simultaneously.

### 4.4.2 Two-phase algorithms

As pointed out by Bent and Van Hentenryck (2006) and Homberger and Gehring (2005), simultaneous reduction of the number of vehicles and total distance travelled in vehicle routing problems by a meta-heuristic controlling a neighborhood search, may lead to an important shortcoming. The objective function often drives the search towards solutions with a small distance. This complicates reaching solutions with a low number of vehicles but higher distance, i.e. the search is mainly guided by the secondary objective. To overcome this shortcoming, a two-phase solution method may be used where the number of vehicles is minimized during the first phase and total distance is minimized during the second phase. Homberger and Gehring (2005) present such a two-phase solution method for the vehicle routing problem with time windows. A  $(\mu, \lambda)$ -evolution strategy is used during the first phase, while a tabu search algorithm is used during the second phase. Bent and Van Hentenryck (2006) use respectively a simulated annealing algorithm and a Large Neighborhood Search (LNS) during their two-phase algorithm for the pickup and delivery problem with time windows. Since these two-phase methods provide very good results, a two-phase solution method for the problem in this paper is proposed as well. During both phases the deterministic annealing meta-heuristic is used.

**First phase: reduce the number of vehicles** Instead of using objective function (1), a specific hierarchical objective function presented by Bent and Van Hentenryck (2006) is used during the first phase of the algorithm. This objective function (10) guides the search towards solutions with a small number of vehicles, while partially ignoring the secondary objective of reducing total distance. Parameter  $l_v$  represents the number of nodes visited by vehicle  $v \in V$ .

$$\text{lex min}(\sum_{i \in N} x_{0i}, -\sum_{v \in V} l_v^2, \sum_{(i,j) \in A} d_{ij}x_{ij} + \sum_{i \in N} d_i) \quad (10)$$

The objective function (10) consists of three hierarchically (or lexicographically) structured objectives. The primary objective is to minimize the number of vehicles used while the secondary objective is to maximize the sum of the squares of the number of nodes in each route. Finally, minimizing total distance is the third objective. The purpose of the second objective is to favour solutions with an unbalanced distribution of nodes over the vehicles over solutions with an even distribution of nodes, i.e. a solution with a few long and a few short routes is preferred over a solution where all routes

have a length close to the average. The idea behind this objective is to remove nodes from shorter routes and insert them into longer routes, thereby gradually reducing the number of vehicles (Bent and Van Hentenryck, 2006).

During each iteration of the deterministic annealing algorithm, all five types of local search operators are applied in a random order. The two route reducing operators have an effect on the primary objective, while the *relocate*, *2-Opt\**, *exchange(2,1)* and *exchange(3,2)* operators improve the secondary objective. The *exchange(1,1)*, *exchange(2,2)* and *exchange(3,3)* operators only have an effect on the third objective and are mainly used to diversify the search. During this phase, the acceptance rule of a neighboring solution is as follows. A new solution is accepted when it is better as the current solution according to objective function (10) or when it has the same number of vehicles and the worsening of the second objective value is smaller than the threshold value  $T$ .

The implementation of the first phase of the sequential algorithm is again straightforward: the initial solution found by the insertion heuristic is improved by the deterministic annealing algorithm for a fixed number of iterations. The implementation for the integrated algorithm is more complex. The insertion heuristic and first phase of the algorithm are not directly applied on the integrated problem. Instead, an initial solution is found in the same way as for the sequential problem (by first determining the optimal empty container allocations by the transportation problem) and during half the number of iterations of the first phase, the number of vehicles is reduced while keeping the empty container allocations fixed. Next, the best solution found so far is transformed to a solution for the integrated problem by relaxing the optimal allocations. During the second half the number of iterations of the first phase, the number of vehicles is reduced further.

**Second phase: reduce total distance** During the second phase, the best solution found during the first phase is further improved with respect to total distance. The original objective function (1) is used. The two route reducing operators are not used since reducing the number of vehicles is not considered anymore.

## 5 Experimental design and results

### 5.1 Experimental design

In order to test the robustness of the algorithm, a  $2^4$  factorial design is set up (Law, 2007). Four problem characteristics are identified. For each characteristic a high (+) and low (-) value is determined. The *time window width* for loaded containers (F1) is a random number between 60 and 120 minutes (-) or between 120 and 240 minutes (+). The *number of container terminals* (F2) is one (-) or three (+). For each instance the *number of nodes* (F3) is 100 (-) or 200 (+). These nodes exist of an even amount of loaded container delivery locations, loaded container pickup locations, empty container supply locations and empty container demand locations. Finally, the *(X,Y)-coordinates* of all nodes (F4) are randomly chosen between 0 and 25 kilometers (-) or between 0 and 50 kilometers (+) on both axes. This results in 16 problem classes as shown in Table 3. For each problem class three random problem instances are generated.

It is assumed that a single vehicle depot is located in the center of the square region and the location of the container terminals is the same for each instance. The planning period equals eight hours. Service time for loading/unloading a container is ten

Table 3: Overview of problem classes

Class	F1	F2	F3	F4	Class	F1	F2	F3	F4
1	-	-	-	-	9	-	-	-	+
2	+	-	-	-	10	+	-	-	+
3	-	+	-	-	11	-	+	-	+
4	+	+	-	-	12	+	+	-	+
5	-	-	+	-	13	-	-	+	+
6	+	-	+	-	14	+	-	+	+
7	-	+	+	-	15	-	+	+	+
8	+	+	+	-	16	+	+	+	+

minutes.

## 5.2 Lower bound

Lower bounds on the sequential and on the integrated problem are found by a time window partitioning method. Time window partitioning or discretization is introduced by Wang and Regan (2002) for solving a  $m$ -TSPTW. The time window of each node is partitioned into smaller parts, which are considered subnodes. Binary flow variables for links between nodes are replaced by binary flow variables for links between subnodes. Exactly one subnode of each node should be visited and for each node a vehicle must enter and leave the same subnode. An underconstrained version of the problem is formulated by replacing constraints (5), (6), (7) and (8) in the formulation of Section 3.1.2 with constraint (11). This underconstrained model results in a lower bound for the original problem.

$$x_{ij}(a_i + s_i + t_{ij} - b_j) \leq 0 \quad \forall (i, j) \in A \quad (11)$$

Wang and Regan (2002) use an iterative approach by partitioning the time windows into smaller parts at each iteration. Recently, Zhang et al (2010) improved this method by immediately looking for a good partitioning width. In this way the underconstrained problem needs to be solved only once, reducing computation time to a large extent. The authors found that instances with up to 800 subnodes could be solved efficiently. In this paper, the method of Zhang et al (2010) is slightly adapted to obtain a better lower bound. The binary restrictions on the flow variables are relaxed. The resulting LP-relaxation of the underconstrained model can be solved very efficiently. Therefore, problems with up to 5000 subnodes can be solved within 5 minutes. The lower bounds generated in this way are tighter than those generated by the integer restricted problems in a comparable amount of computation time.

Separate lower bounds are calculated for the sequential and integrated problem. To obtain a lower bound on the number of vehicles, the objective function of the problem is changed to minimizing the total time needed to perform all transport requests. The total time needed is then divided by the length of the planning period to find a lower bound on the number of vehicles (Koo et al, 2004). A lower bound on the total distance is found by removing the fixed vehicle costs from the objective function. It should be noted that lower bounds on the integrated problem are less tight than those of the sequential problem. The reason is that the average time window width is much larger



Table 4: Overview of results

Value	Sequential		Integrated	
	Single phase	Two-phase	Single phase	Two-phase
Average number of vehicles	10.46	10.04	10.33	9.99
Average gap (absolute)	1.00	0.58	1.08	0.74
Average distance (km)	1813	1837	1805	1815
Average gap (%)	2.78	3.74	4.74	5.33
Average computation time (s)	3.40	5.38	4.27	6.59

for the integrated problem. Therefore the partitioning width was chosen somewhat wider in order to ensure a lower bound is found within reasonable computation time.

### 5.3 Results

The best parameter values are found by testing the algorithms for a single problem instance of each class. Best results are achieved with the following parameter values. The insertion heuristic is iterated 1000 times to obtain a good starting solution. The deterministic algorithm is run for 50000 iterations (during each phase). For the single phase algorithms and the second phase of the two-phase algorithms, the maximum threshold value  $T_{max}$  is set at 4 kilometers for 25 square kilometer regions and at 8 kilometers for 50 square kilometer regions. The threshold reduction parameter  $\Delta T$  is set at  $T_{max}/2500$ . During the first phase of the two-phase algorithms the maximum threshold value  $T_{max}$  is set at 6 for the sequential algorithm and at 12 for the integrated algorithm while the threshold reduction parameter  $\Delta T$  is set at  $T_{max}/1500$ . For all algorithms, the reset parameter  $n_{imp}$  is equal to five times the number of vehicles used in the current best solution. The transportation problem and deterministic annealing algorithm are implemented in C on a 2.1GHz Intel Core 2 laptop with 4GB RAM. The lower bounds are computed using Lingo 10.0.

All four algorithms are tested on the 48 problem instances. For each instance, average results over five runs of the algorithm are obtained. Detailed results are shown in Appendix A. An overview of results is shown in Table 4. The first two rows in Table 4 show the average number of vehicles used and the average absolute gap with the corresponding lower bound. The average distance travelled and the average relative gap with the corresponding lower bound, are presented in rows three and four. In the last row, average computation in seconds is shown. From Table 4, it can be concluded that all algorithms are able to find good quality solutions in a small amount of time. A comparison of the sequential and integrated approach should be based on real values instead of the gaps with the lower bounds since different bounds are calculated for both problems. Results show that on average the integrated approach offers better results than the sequential approach, both in terms of the number of vehicles and total distance. Comparing results of each problem instance individually shows that the integrated approach almost always performs better (21%) or equally good (74%) in terms of the number of vehicles used. When both approaches result in the same number of vehicles, the integrated approach is able to find a better solution in terms of distance in 95% of the cases.

It should be noted that although the computation time of the algorithms is very small, the average computation time of the integrated approach is slightly higher than

that of the sequential approach. To further substantiate the conclusion that the integrated approach performs best, the computational experiments are redone for the sequential algorithms. This time, instead of a fixed number of iterations, the sequential algorithms are run for the same time as the computation time needed by the integrated approach. However, the results of the sequential approach hardly improve (same number of vehicles and average distances of 1812 and 1834 kilometers respectively for the single and two-phase method) and thus the integrated approach still provides better results.

When comparing the results of the single and two-phase algorithms, it is clear that the two-phase method performs much better in terms of the number of vehicles. Average distances are slightly higher for the two-phase algorithms than for the single phase algorithms which can be explained by the fact that the two objectives (minimize vehicles and distance) are often conflicting. Since minimizing the number of vehicles is prioritized over minimizing total distance, it can be concluded that best solutions are obtained by the integrated two-phase algorithm. This corresponds with the findings of Homberger and Gehring (2005) and Bent and Van Hentenryck (2006) that two-phase methods work well on problems with a hierarchical objective function.

## 5.4 Comparison with previous results

To further assess the quality of the proposed algorithms in this paper, a comparison is made with previous results obtained by a time window partitioning method on a similar problem (Zhang et al, 2010). Since minimizing the number of vehicles used is not an objective in Zhang et al (2010), a comparison is made with the integrated single phase algorithm presented in this paper. The objective function of our algorithm is changed to minimize the total operation time of the vehicles. To meet the assumption of Zhang et al (2010) that there are multiple vehicle depots with a limited number of vehicles, the travel times between the vehicle depots and the nodes are initially set to zero. After a solution has been found by the algorithm, an optimal allocation of vehicles to the routes is made by solving a small transportation problem. Obviously, the algorithm could be improved by addressing this decision during the search. For example, a local search operator that swaps the starting depots of two routes could be introduced.

In Table 5, average results obtained by the integrated single phase algorithm on the twenty instances of Zhang et al (2010) are presented and are compared with the results of the original paper. Average results for instances with the same time window width are presented. For both methods, the average number of vehicles, the average gap between the total operation time and the lower bound of Zhang et al (2010) and the computation time are shown. Results in Table 5 show that the integrated single phase algorithm presented in this paper outperforms the time window partitioning method described by Zhang et al (2010). Besides, computation time is small.

## 6 Conclusions and future research

Drayage operations often constitute a large part of total costs of an intermodal transport. In this paper, a sequential and an integrated approach are proposed to plan loaded and empty container drayage operations. For both approaches, a single and a two-phase deterministic annealing algorithm are presented to solve a full truckload routing problem with time windows. It can be concluded that for both solution approaches good

Table 5: Comparison of results with Zhang et al (2010)

TW width	Zhang et al (2010)			Own results		
	Vehicles	Gap (%)	Time (s)	Vehicles	Gap (%)	Time (s)
1h	47.6	1.20	103	47.1	1.13	3
2h	48.2	2.68	89	47.2	1.90	3
3h	43.2	3.96	158	40.9	2.69	3
4h	37.4	4.83	124	34.6	4.26	3
Average	44.1	3.17	118	42.5	2.49	3

quality solutions can be obtained in a small amount of time. Results clearly indicate that the integrated approach performs better than the sequential approach. Besides, results in this paper confirm the advantage of using a two-phase approach for vehicle routing problems with a hierarchical objective function. Finally, it is shown that the proposed integrated solution method improves previous results on a similar problem.

Future research could focus on a number of extensions of the problem addressed in this paper. The number of empty containers available at each terminal could be assumed to be limited. This strongly complicates the problem as is shown in Zhang et al (2011b). Another extension may be the problem where for some containers and/or vehicles, two containers can be transported simultaneously by a single vehicle. Finally, the hierarchical objective function may be replaced by a multi-objective function. In this way the trade-off between the number of vehicles used and total distance travelled could be explored.

## A Detailed results

Table A.1 shows detailed results for the experiments described in Section 5.3.

## References

- Baldacci R, Bodin L, Mingozzi A (2006) The multiple disposal facilities and multiple inventory locations rollon-rolloff vehicle routing problem. *Comput Oper Res* 33(9):2667–2702
- Bent R, Van Hentenryck P (2006) A two-phase hybrid algorithm for pickup and delivery vehicle routing problems with time windows. *Comput Oper Res* 33(4):875–893
- Braekers K, Janssens GK, Caris A (2010) A deterministic annealing algorithm for simultaneous routing of loaded and empty containers. In: *Proceedings of the Industrial Simulation Conference, Budapest, Hungary*, pp 172–176
- Braekers K, Janssens GK, Caris A (2011) Challenges in managing empty container movements at multiple planning levels. *Transport Rev* 31(6):681–708
- Bräysy O, Berger J, Barkaoui M, Dullaert W (2003) A threshold accepting meta-heuristic for the vehicle routing problem with time windows. *Cent Eur J Oper Res* 11(4):369–387

Table A.1: Detailed results

Instance	Sequential				Integrated			
	Single phase		Two-phase		Single phase		Two-phase	
	V <sup>a</sup>	D <sup>b</sup>	V	D	V	D	V	D
1.1	7	1032	6.4	1063	7	1019	6	1066
1.2	7	968	6.4	999	6.6	973	6	994
1.3	6	984	6	989	6.4	963	6	971
2.1	6	1012	6	1013	6	1001	6	1003
2.2	6	983	6	987	6	988	6	974
2.3	6	928	6	927	6	923	6	919
3.1	6	830	6	830	6	812	6	811
3.2	6	760	6	755	6	743	6	750
3.3	6	728	6	727	6	701	6	702
4.1	6	702	6	705	6	697	6	696
4.2	6	689	6	689	6	682	6	680
4.3	6	719	6	723	6	699	6	697
5.1	12.8	1789	12	1820	12	1803	12	1800
5.2	12	1837	12	1833	12	1827	12	1823
5.3	12	1654	11	1712	12	1651	11	1697
6.1	12	1645	11	1685	12	1659	11	1680
6.2	12	1661	11	1696	11.6	1660	11	1671
6.3	12	1795	11	1837	12	1793	11.2	1809
7.1	11	1436	11	1424	11	1403	11	1403
7.2	11	1315	11	1315	11	1283	10.8	1299
7.3	11	1302	10	1342	11	1301	10	1322
8.1	11	1361	11	1363	11	1346	11	1339
8.2	11	1306	10.4	1346	11	1292	10.4	1324
8.3	10.4	1175	10	1185	10	1163	10	1162
9.1	9	1936	9	1939	9	1935	9	1930
9.2	9	1986	9	1986	9	1977	9	1959
9.3	9	1826	8	1884	8.2	1836	8	1851
10.1	9	1943	8	1987	9	1941	8	1966
10.2	9	1845	8	1926	8.6	1865	8	1891
10.3	8.4	1894	8	1896	8.6	1881	8	1883
11.1	8	1609	8	1613	8	1610	8	1607
11.2	8	1683	8	1683	8	1666	8	1669
11.3	9	1505	8.6	1567	8.8	1516	8	1582
12.1	8	1478	7	1555	8	1455	7	1528
12.2	8	1564	8	1581	8	1553	8	1556
12.3	7	1354	7	1362	7	1354	7	1355
13.1	17	3381	16	3435	16	3357	15.4	3354
13.2	17	4068	17	4074	17	4047	17	4045
13.3	17	3501	16	3537	16.4	3503	16	3519
14.1	16	3345	15.2	3436	16	3344	15	3408
14.2	17	3595	16	3647	16	3637	16	3622
14.3	16	3231	15	3305	15.8	3279	15	3265
15.1	15	2948	14.8	2947	15	2908	15	2921
15.2	15	2810	14	2855	14.8	2803	14	2824
15.3	16	2731	15	2770	15.2	2722	14.8	2744
16.1	14.4	2802	14	2820	14.6	2730	14	2739
16.2	14	2681	14	2678	14	2652	14	2643
16.3	14	2706	14	2713	14	2681	14	2678

<sup>a</sup> V: average number of vehicles used (over five runs)<sup>b</sup> D: average total distance travelled (over five runs)

- Bräysy O, Dullaert W, Hasle G, Mester D, Gendreau M (2008) An effective multi-start deterministic annealing metaheuristic for the fleet size and mix vehicle routing problem with time windows. *Transport Sci* 42(3):371–386
- Caris A, Janssens GK (2009) A local search heuristic for the pre- and end-haulage of intermodal container terminals. *Comput Oper Res* 36(10):2763–2772
- Caris A, Janssens GK (2010) A deterministic annealing algorithm for the pre- and end-haulage of intermodal container terminals. *Int J Comput Aided Eng Tech* 2(4):340–355
- Chang H, Jula H, Chassiakos A, Ioannou P (2008) A heuristic solution for the empty container substitution problem. *Transport Res E-Log* 44(2):203–216
- Chu Q (1995) Dynamic and stochastic models for container allocation. PhD thesis, Department of Ocean Engineering, Massachusetts Institute of Technology
- Cordeau JF, Laporte G, Savelsbergh MWP, Vigo D (2007) Vehicle routing. In: Barnhart C, Laporte G (eds) *Transportation, Handbooks in Operations Research and Management Science*, vol 14, Elsevier, Amsterdam, pp 367–428
- Crainic TG, Gendreau M, Dejax P (1993) Dynamic and stochastic models for the allocation of empty containers. *Oper Res* 41(1):102–126
- Currie RH, Salhi S (2004) A tabu search heuristic for a full-load, multi-terminal, vehicle scheduling problem with backhauling and time windows. *J Math Model Algorithm* 3(3):225–243
- Deidda L, Di Francesco M, Olivo A, Zuddas P (2008) Implementing the street-turn strategy by an optimization model. *Marit Pol Manag* 35(5):503–516
- Dejax PJ, Crainic TG (1987) A review of empty flows and fleet management models in freight transportation. *Transport Sci* 21(4):227–248
- Desaulniers G, Lavigne J, Soumis F (1998) Multi-depot vehicle scheduling problems with time windows and waiting costs. *Eur J Oper Res* 111(3):479–494
- Desrosiers J, Sauvé M, Soumis F (1988) Lagrangian relaxation methods for solving the minimum fleet size multiple traveling salesman problem with time windows. *Manage Sci* 34(8):1005–1022
- Di Francesco M, Manca A, Olivo A, Zuddas P (2006) Optimal management of heterogeneous fleets of empty containers. In: *Proceedings of the International Conference on Information Systems, Logistics and Supply Chain*, Lyon, France, pp 922–931
- Dueck G, Scheuer T (1990) Threshold accepting: A general purpose optimization algorithm appearing superior to simulated annealing. *J Comput Phys* 90(1):161–175
- Erera AL, Smilowitz K (2008) Intermodal drayage routing and scheduling. In: Ioannou P (ed) *Intelligent Freight Transportation, Automation and Control Engineering Series*, CRC Press, Boca Raton, FL, pp 171–188
- Escudero A, Muñozuri J, Arango C, Onieva L (2011) A satellite navigation system to improve the management of intermodal drayage. *Adv Eng Inform* 25(3):427–434

- Ford LR, Fulkerson DR (1956) Solving the transportation problem. *Manage Sci* 3(1):24–32
- Francis P, Zhang G, Smilowitz K (2007) Improved modeling and solution methods for the multi-resource routing problem. *Eur J Oper Res* 180(3):1045–1059
- Gronalt M, Hirsch P (2007) Log-truck scheduling with a tabu search strategy. In: Doerner K, Gendreau M, Greistorfer P, Guthjahr W, Hartl R, Reimann M (eds) *Metaheuristics - Progress in Complex Systems Optimization*, Springer, New York, NY, pp 64–88
- Gronalt M, Hartl RF, Reimann M (2003) New savings based algorithms for time constrained pickup and delivery of full truckloads. *Eur J Oper Res* 151(3):520–535
- Hadjar A, Soumis F (2009) Dynamic window reduction for the multiple depot vehicle scheduling problem with time windows. *Comput Oper Res* 36(7):2160–2172
- Homberger J, Gehring H (2005) A two-phase hybrid metaheuristic for the vehicle routing problem with time windows. *Eur J Oper Res* 162(1):220–238
- Huth T, Mattfeld DC (2009) Integration of vehicle routing and resource allocation in a dynamic logistics network. *Transport Res C-Emer* 17(2):149–162
- Ileri Y, Bazaraa M, Gifford T, Nemhauser G, Sokol J, Wikum E (2006) An optimization approach for planning daily drayage operations. *Cent Eur J Oper Res* 14(2):141–156
- Imai A, Nishimura E, Current J (2007) A Lagrangian relaxation-based heuristic for the vehicle routing with full container load. *Eur J Oper Res* 176(1):87–105
- Jula H, Chassiakos A, Ioannou P (2003) Empty container interchange report: Methods for modeling and routing of empty containers in the Los Angeles and Long Beach port area. Final report, Center for the Commercial Deployment of Transportation Technologies, California State University, Long Beach, CA
- Jula H, Dessouky M, Ioannou P, Chassiakos A (2005) Container movement by trucks in metropolitan networks: modeling and optimization. *Transport Res E-Log* 41(3):235–259
- Koo PH, Lee WS, Jang DW (2004) Fleet sizing and vehicle routing for container transportation in a static environment. *OR Spectrum* 26(2):193–209
- Law A (2007) *Simulation modeling and analysis*, 4th edn. McGraw-Hill, New York, NY
- Mes M, van der Heijden M, van Harten A (2007) Comparison of agent-based scheduling to look-ahead heuristics for real-time transportation problems. *Eur J Oper Res* 181(1):59–75
- Mes M, van der Heijden M, Schuur P (2010) Look-ahead strategies for dynamic pickup and delivery problems. *OR Spectrum* 32(2):395–421
- Mingozzi A, Bianco L, Ricciardelli S (1995) An exact algorithm for combining vehicle trips. In: Daduna J, Branco I, Paixao J (eds) *Computer-aided Transit Scheduling*, *Lecture Notes in Economics and Mathematical Systems* 430, Springer, Berlin, pp 145–172

- Mitrović-Minić S, Krishnamurti R (2006) The multiple tsp with time windows: vehicle bounds based on precedence graphs. *Oper Res Lett* 34(1):111–120
- Namboothiri R, Erera AL (2008) Planning local container drayage operations given a port access appointment system. *Transport Res E-Log* 44(2):185–202
- Nikolakopoulos A, Sarimveis H (2007) A threshold accepting heuristic with intense local search for the solution of special instances of the traveling salesman problem. *Eur J Oper Res* 177(3):1911–1929
- Olivo A, Zuddas P, Di Francesco M, Manca A (2005) An operational model for empty container management. *Marit Pol Manag* 7(3):199–222
- Palmgren M, Rönnqvist M, Värbrand P (2003) A solution approach for log truck scheduling based on composite pricing and branch and bound. *Int Trans Oper Res* 10(5):433–447
- Potvin JY, Rousseau JM (1995) An exchange heuristic for routeing problems with time windows. *J Oper Res Soc* 46(12):1433–1446
- Smilowitz K (2006) Multi-resource routing with flexible tasks: an application in drayage operations. *IIE Trans* 38(7):577–568
- Tarantilis C, Kiranoudis C, Vassiliadis V (2004) A threshold accepting metaheuristic for the heterogeneous fixed fleet vehicle routing problem. *Eur J Oper Res* 152(1):148–158
- Wang X, Regan AC (2002) Local truckload pickup and delivery with hard time window constraints. *Transport Res B-Meth* 36(2):97–112
- Zhang G, Smilowitz K, Erera A (2011a) Dynamic planning for urban drayage operations. *Transport Res E-Log* 47(5):764–777
- Zhang R, Yun WY, Moon IK (2009) A reactive tabu search algorithm for the multi-depot container truck transportation problem. *Transport Res E-Log* 45(6):904–914
- Zhang R, Yun WY, Kopfer H (2010) Heuristic-based truck scheduling for inland container transportation. *OR Spectrum* 32(3):787–808
- Zhang R, Yun WY, Moon IK (2011b) Modeling and optimization of a container drayage problem with resource constraints. *Int J Prod Econ* 133(1):351–359