Published in: Networks (2017), Volume 69, Issue 3, Pages 304 - 3161
DOI: 10.1002/net.21738
Status : Postprint (Author's version)

Iterated Local Search for the Capacitated Vehicle Routing Problem with Sequence-Based Pallet Loading and Axle Weight Constraints

Hanne Pollaris, Kris Braekers, An Caris, and Gerrit K. Janssens
UHasselt, Research group Logistics, Agoralaan, Diepenbeek 3590, Belgium

Kris Braekers
Research Foundation Flanders (FWO), Egmontstraat 5, Brussels BE-1000, Belgium

Sabine Limbourg
HEC Management School, University of Liege (ULg), QuantOM, Rue Louvrex 14 (N1), Liege 4000, Belgium

KEYWORDS: Vehicle Routing Problem; axle weight constraints; routing & loading; iterated local search; sequence-based loading; Capacitated Vehicle Routing Problem

ABSTRACT

In this article an Iterated Local Search algorithm for the capacitated vehicle routing problem with sequence-based pallet loading and axle weight constraints is presented. Axle weight limits impose a great challenge for transportation companies. Yet, the literature on the incorporation of axle weight constraints in vehicle routing models is very scarce. The effect of introducing axle weight constraints in a CVRP on total routing cost is analyzed. Results show that integrating axle weight constraints does not lead to a large cost increase. However, not including axle weight constraints in the planning profcess may induce major axle weight violations.

1. INTRODUCTION

The capacitated vehicle routing problem with sequence-based pallet loading and axle weight constraints is an extension of the classical Capacitated Vehicle Routing Problem (CVRP). It integrates loading constraints in a routing problem and is based on a real-world transportation problem. For state-of-the-art reviews of the literature concerning the combination of Vehicle Routing Problems (VRP) and loading problems, the reader is referred to Iori and Martello [12] and Pollaris et al. [22]. Vehicle routing problems consider the distribution of goods between depots and customers or nodes [26]. The goal is to find a set of routes for a fleet of vehicles which fulfills every customer demand and where the objective function (e.g., total distance, routing costs) is optimized. The basic version of the vehicle routing problem is the CVRP. The CVRP considers a homogeneous vehicle fleet with a fixed capacity (in terms of weight or number of items) which delivers goods from a depot to customer locations. Split deliveries are not allowed. In this article, a variant of the classical CVRP is considered. The demand of the customers consists of pallets. These pallets may be placed in two rows inside the vehicle but cannot be stacked on top of each other because of their weight, fragility, or customer preferences. Sequence-based loading is imposed which ensures that when arriving at a customer, no pallets belonging to customers served later on the route block the removal of the pallets of the current customer. Furthermore, the capacity of a truck is not only expressed in total weight and number of pallets but also consists of a maximum weight on the axles of the truck. Axle weight limits pose a challenge to transportation companies as they incur high fines in the event of non-compliance. Weigh-In-Motion (WIM) systems on high- ways monitor axle weight violations of trucks while driving which increases the probability that axle weight violations are detected [13]. Furthermore, trucks with overloaded axles represent a threat for traffic safety and may cause serious damage to the road surface. To our knowledge, Lim et al. [14] and Alonso et al. [2] are the only authors that address axle weight constraints in a container loading problem. Lim et al. [14] develop a heuristic method to tackle the single container loading problem with axle weight constraints. Alonso et al. [2] develop integer linear programming models to tackle multicontainer loading problems with axle weight constraints in which items are first packed on pallets and afterward, pallets are placed onto trucks.

The CVRP with sequence-based pallet loading and axle weight constraints was introduced in Pollaris et al. [23]. A Mixed Integer Linear Programming model (MILP) is proposed to solve the problem to optimality for networks of up to 20 nodes. To the best of our knowledge, this is currently the only paper that addresses the integration of axle weight constraints in a VRP. The problem has similarities with the Multi-Pile VRP (MP-VRP), the Double Traveling Salesman Problem with Multiple Stacks (DTSPMS), and the Traveling Salesman Pickup and Delivery Problem (TSPPD) with multiple stacks. Doerner et al. [8] develop a Tabu Search (TS) method and an Ant Colony Optimization (ACO) method to solve the MP-VRP, based on a real-world application regarding the transport of wooden chipboards. For every order, chipboards of the same type (small or large) are grouped into a unique item, which is placed onto a single pallet. The vehicle is divided into three piles on which pallets can be stacked. Pallets containing large chipboards can extend over multiple piles. The other pallets can be placed into a single pile. Because of this specific configuration of pallets placed into multiple piles, the original three-dimensional problem can be reduced to a one-dimensional one. Tricoire et al. [27] develop a combination of VNS and branch-and-cut to solve the MP-VRP exactly for small-size instances and heuristically for large-size instances. In both papers, sequence-based loading is taken into account. The DTSPMS, proposed by Petersen and Madsen [21], considers pickup and delivery of goods performed in two separate networks in vehicles with multiple stacks. All pickups must be made before any delivery can take place. The goods cannot be repacked, nor vertically stacked. The goods can be placed in several rows (horizontal stacks). In each row, sequence-based loading (which is equivalent to Last-In-First-Out as only a single dimension is considered) is assumed. It is assumed that each order consists of a single item. The problem is based on a real-world application in which in a first phase a container is loaded onto a truck to perform pickup operations and returned by that truck to a depot or terminal. In a second phase, the container is loaded onto a train, ship, plane, or another truck and transported to another depot or terminal. In the depots or terminals, there are no facilities to repack the items inside the container. In the final phase, the container is again transferred to a truck which performs the delivery operations [21]. Petersen and Madsen [21], Felipe et al. [11], and Felipe et al. [10] develop heuristic

Published in: Networks (2017), Volume 69, Issue 3, Pages 304 - 3161
DOI: 10.1002/net.21738
Status : Postprint (Author's version)

methods to solve the DTSPMS, while Lusby et al. [18], Petersen et al. [20], Lusby and Larsen [17], Alba et al. [1], and Alba et al. [3] propose exact algorithms. Côté et al. [6] and Côté et al. [5] consider the TSPPD with multiple stacks with LIFO loading. They propose a heuristic method and a branch-and-cut algorithm, respectively. Øvstebø et al. [19] examine a similar problem on Roll-on/Roll-of (RoRo) ships that transport cargo on wheels. The decks on the ship may be divided into lanes in which the cargo may be placed. The lanes may be compared to stacks in a truck. Sequence-based loading is considered as a soft constraint. A penalty cost is incurred if the constraint is violated.

In this article, we present a metaheuristic for the CVRP with sequence-based pallet loading and axle weight constraints and compare the results to those of the CVRP without axle weight constraints. To the best of our knowledge, it is the first time that a vehicle routing problem with axle weight constraints is studied on networks of realistic size. More- over, we present the first heuristic solution approach for this problem. The goal of this article is twofold. First, the performance of the metaheuristic is validated by comparing results with those from Pollaris et al. [23]. Second, instances with networks of 50-100 customers are analysed to observe the extent to which axle weight limits are violated when ignored in the planning process and the necessary additional costs to avoid these violations.

Figure 1 indicates the effect that axle weight constraints may have on a routing solution by means ofa simple example (see Pollaris et al. [23] for more details). Consider a single truck with a capacity of 22 pallets. Sequence-based load- ing is assumed and the vehicle is unloaded through the rear. Suppose four customers, each with a demand of five pal- lets, need to be delivered by this vehicle from a single depot. Total weight of the five pallets is 12, 2, 2, and 12 tonnes, respectively. Figure 1a shows the shortest route and the corresponding loading scheme when axle weight constraints are ignored. This solution may no longer be feasible when axle weight constraints are accounted for, because the relatively large weight of the pallets of customer 4 is mainly carried by the axles of the tractor, which typically have the lowest net weight capacity. The optimal route when axle weight constraints are accounted for many, therefore, change to the one depicted in Figure 1b, in which the weight of these pallets is distributed more evenly between both axles. However, this clearly results in an increase in optimal route length.
The remainder of this article is organized as follows. In the next section, a problem description is given for the CVRP with sequence-based pallet loading and axle weight constraints. Section 3 describes an Iterated Local Search method (ILS) method which is developed to tackle the problem. In section 4, computational experiments of the ILS are described and a comparison is made between the CVRP with and with- out axle weight constraints. In the final section, conclusions and future research opportunities are discussed (section 5).

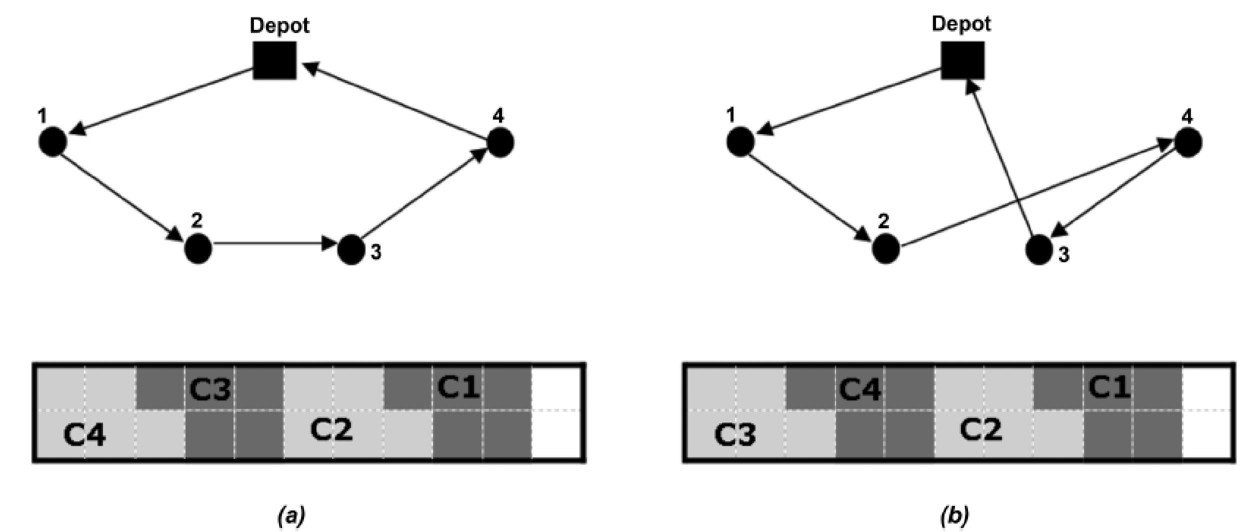
2. PROBLEM DESCRIPTION

The problem addressed in this article is the CVRP with sequence-based pallet loading and axle weight constraints. For a mathematical formulation of the problem, the reader is referred to Pollaris et al. [23]. The goal is to find a vehicle routing plan such that the demand of each customer is satisfied and the total distance traveled is minimized. Demand of the customers consists of europallets (80 x 120 cm). These pallets are delivered from a single depot with an unlimited fleet of homogeneous vehicles. It is assumed that all pallets of a single customer have the same weight and that the weight is uniformly distributed inside each pallet, that is, the center of gravity of a pallet lies in its geometric midpoint. Pallets may be placed in two horizontal rows in the truck. Pallets are loaded and unloaded through the rear of the vehicle. Sequence-based loading is assumed. Pallets are packed dense inside the vehicle. This means that there may be no gap between two consecutive pallets in the container and that all pallets are alternately packed in the left and right row. Furthermore, dense packing entails that there may not be an open space between the front of the container and the first pallets that are packed. Dense packing is often imposed to increase the stability of the load as it restricts the moving area of the pallets considerably. The driver therefore needs to spend less time on securing the cargo than when pallets are spread over the vehicle. Vertical stacking is not allowed due to fragility of goods. Moreover, customers usually do not want goods of other customers to be stacked on top of their goods.

Axle weight is the weight that is placed on the axles of the truck. A truck with five axles is illustrated in Figure 2. The first axle, also called the *steering axle*, and the second axle, called the *driving axle*, both belong to the tractor. The axles of the trailer are *tridem axles*. Tridem axles are three successive axles with a distance of less than 1.8 and more than 1 m between the middle of the first axle and the middle of the second axle, and between the middle of the second axle and the middle of the third axle. When item j is placed in a vehicle, the weight of the item is divided over the axles of the tractor and the axles of the trailer. Variable a_j^F represents the weight of the items of node j placed on the coupling of the truck (which is the link between the tractor and the trailer). The weight on the coupling is carried by the axles of the tractor. Variable a_j^R represents the weight of the items of node j on the axles of the trailer. As weight distribution varies with every pickup or delivery, this should be monitored not just at the point of departure but throughout the journey. A load that is placed at the rear of the vehicle (behind the axles of the trailer), has a negative weight on the axles of the tractor. For this reason, it is possible that by unloading this item a violation of the weight limits of the axles of the tractor is induced.

Published in: Networks (2017), Volume 69, Issue 3, Pages 304 - 3161
DOI: 10.1002/net.21738
Status : Postprint (Author's version)

Figure 1. *Graphical representation of an optimal vehicle route and the corresponding loading scheme of a container (in top view) (a) without axle weight constraints, (b) with axle weight constraints. The load of, respectively, customer 1, 2, 3, and 4 is indicated by C1, C2, C3, and C4.*



The calculation of the weight of the pallets of customer *j* on the coupling point or the axles of the tractor (a_j^F) and on the axles the trailer (a_j^R) is presented in Equations (1) and (2). Figure 3 graphically presents the parameters in Equations (1) and (2). The weight of the pallets of customer *j* is denoted by *wj*. *CGj* represents the distance from the front of the container to the center of gravity of the pallet of customer *j*. Parameter *c* denotes the distance from the front of the container to the coupling. The final parameter *d* represents the distance between the coupling and the central axle of the trailer.

$$a_j^R = \frac{(CG_j - c)}{d} w_j \qquad (1)$$

$$a_j^F = w_j - a_j^R \qquad (2)$$

The weight of the pallets is divided over the axles of the trailer and the axles of the tractor. The distribution of the weight over the axles depends on the distance between the pallet and the axles. The first factor of the second member in Equation (1) $(\frac{(CG_j - c)}{d})$ computes the percentage of the weight that is assigned to the axles of the trailer by dividing the distance between the coupling and the center of gravity of the item by the distance between the coupling and the central axle of the trailer. The second factor is the weight of the item. The larger the distance between the item and the coupling, the higher the percentage of weight that is distributed to the axles of the trailer will be. The weight on the coupling is computed in Equation (2) by subtracting the weight on the axles of the trailer from the weight of the item.

The values of the upper bounds of the weight on the axles of the tractor and on the axles of the trailer depend on vehicle characteristics and are specified in legislation. The lower bound of the weight on the axles of the tractor may also be fixed in legislation. In this article, we take the Belgian legislation as an example. Belgian legislation (KB 15.03.1968 art 32 bis) incorporated European Directive 97/27/EC that specifies that the mass corresponding to the load on the driving axle must be at least 25? of the total mass of the loaded truck. There are no specific guidelines concerning the lower bound on the weight on the axles of the trailer except for the fact that it can never be negative as a negative weight on one of the axles would cause the truck to overturn. Therefore, the weight of the pallets that is placed on the axles of the trailer added up with the weight of the empty truck that is carried by the axles of the trailer should be higher than or equal to zero. Similarly, the weight of the pallets that is placed on the axles of the tractor added up with the weight of the empty truck that is carried by the axles of the tractor should be higher than or equal to zero. The lower and upper bounds refer to the total weight on the axles, which is the sum of the axle weights of each individual customer.

Figure 2*. Axle weight tractor (steering axle, driving axle) and trailer (tridem axles) (figure adapted from TruckScience).*

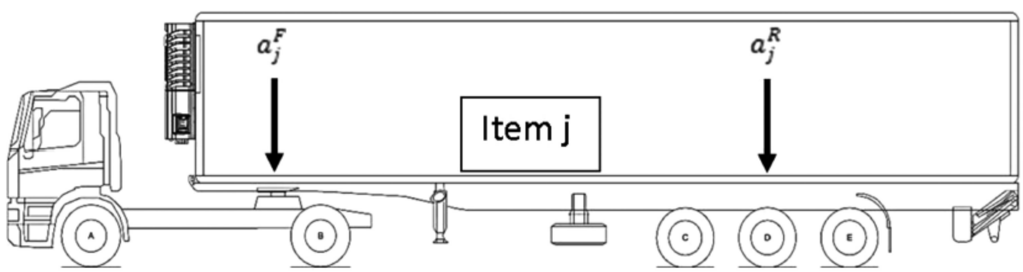
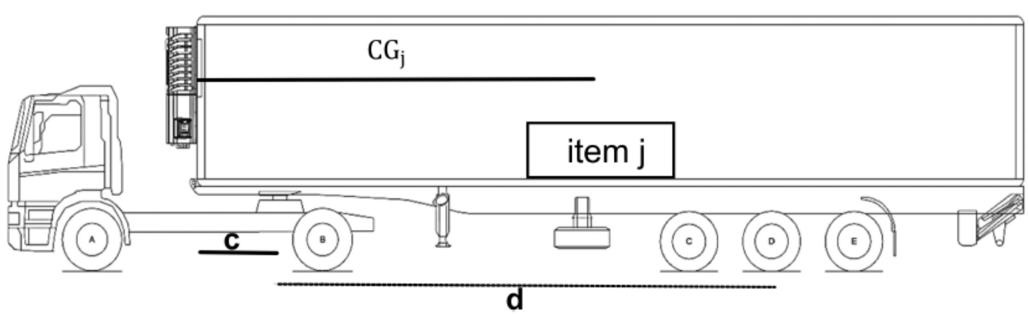


Figure 3. *Tractor (with two axles) and trailer (with tridem axles) (figure adapted from TruckScience).*



3. SOLUTION METHOD

The proposed solution method is based on an Iterated Local Search (ILS) framework which is proven to be a highly effective heuristic for routing problems [16]. The ILS consists of four procedures (Generate initial solution, Local Search, Perturbation, Acceptance Criterion). The procedures are presented in Algorithm 1. First, an initial solution is constructed. This solution is improved using local search until a local optimum is reached. The local search is performed by a Variable Neighborhood Descent (VND). A

Published in: Networks (2017), Volume 69, Issue 3, Pages 304 - 3161
DOI: 10.1002/net.21738
Status : Postprint (Author's version)

new starting point for the local search is generated by perturbing the current solution. The acceptance criterion determines after the local search with which solution the process continues. The ILS stops after *α* consecutive non-improving iterations. A non-improving iteration is an iteration in which no new best solution was found. For more information regarding the general ILS framework, the reader is referred to Lourenço et al. [16]. Note that as the local search is performed by a VND, the algorithm may also be called Iterated Variable Neighborhood Descent, as used in Chen et al. [4].

Algorithm 1 *Steps of the ILS*

Initialization

1: s^o ← Generate initial solution
2: s, s^b ← Local search on s^o
3: **repeat**
  s ← Perturbation on s
  s ← Local search on s
  s, s^b ← Acceptance criterion
4: **until** $non\_improving\_it > \alpha$

To improve the efficiency of the algorithm in this article, a pool of feasible and infeasible routes is constructed. Each time a route is proven to be (in)feasible in terms of loading, this route is stored in the appropriate pool. This avoids duplicate loading feasibility checks of a single route. In the following sections, our implementation of the ILS is described.

3.1. INITIAL SOLUTION

Routes are constructed by inserting nodes one by one. The load of a node that is inserted is placed directly behind the load of previously inserted nodes. To obtain a feasible initial solution, special attention is given to the insertion of *difficult* nodes. Nodes are considered *difficult* if they cannot be inserted feasibly in the front of a truck because the mass of the pallets exceeds the capacity of the axles of the tractor. These axles typically have the lowest axle weight capacity. The load of those nodes should therefore be placed more toward the end of the truck. As sequence-based loading is assumed and no gaps are allowed between the front of the truck and the load (see section 2), these nodes can only be feasibly inserted after the insertion of one or several other (*non-difficult*) nodes in a route.

For each *difficult* node, a list is constructed with all nodes or combinations of two nodes that would lead to a feasible packing scheme when these nodes precede the *difficult* node. To decide for each *difficult* node which option is chosen, a binary constraint satisfaction problem (BCSP) is solved. The following notation is used:

Ω = set of *difficult* nodes (index *j*)

Φ= set of *non - difficult* nodes (index *i*)

ψ_j = set of options for *difficult* node *j (*index *a)*

$$k^j_{ai} = \begin{cases} 1 & \text{if } \textit{non-difficult} \text{ node } i \text{ belongs to option } a \\ & \quad \text{of } \textit{difficult} \text{ node } j \\ 0 & \text{otherwise} \end{cases}$$

The decision variables are defined as:

$$x_{aj} = \begin{cases} 1 & \text{if option } a \text{ is chosen for } \textit{difficult} \text{ node } j \\ 0 & \text{otherwise} \end{cases}$$

The constraints are as follows:

$$\sum_{a \in \Psi_j} x_{aj} = 1 \qquad \forall j \in \Omega \qquad (3)$$

$$\sum_{j \in \omega} \sum_{a \in \Psi_j} k^j_{ai} \cdot x_{ij} \leq 1 \qquad \forall i \in \Phi \qquad (4)$$

No objective function is specified as the only goal is to find a solution that meets all constraints. Constraint (3) ensures that for every *difficult* node, a single option is chosen. Constraint (4) makes sure that each *non-difficult* node *i* can only be inserted once. The BCSP is solved with CPLEX 12.6 with the default parameters. Preliminary tests (see section 4.2) have shown that a solution can often already be obtained when allowing only a single *non-difficult* node to precede each *difficult* node. Additionally, for some instances, considering both a single and a combination of two *non-difficult* nodes, considerably increases computation time. Therefore, the BCSP is first solved with options consisting of a single *non-difficult* node only. When the BCSP is not able to not find a feasible solution, a combination of two nodes is allowed.

When the BCSP finds a feasible solution, each *difficult* node is inserted into a route along with the node(s) from the corresponding option that was selected by the BCSP.
Therefore, as many routes as *difficult* nodes are created. The remaining *non-difficult* nodes are inserted with a regret-2 insertion heuristic. The regret value of a node is defined as the absolute difference in costs between the cheapest insertion of a node and the second cheapest insertion of that node into another route. In each iteration, insertions in the existing routes are considered as well as in an additional empty route. The node with the highest regret value is inserted in its best insertion position. This procedure continues until all nodes are feasibly inserted into a route.

3.2. Local Search

We apply a local search in which four neighborhoods are used. The exchange operator ([28]) swaps two nodes which can be either from the same route or from different routes. The 2-opt operator ([7]) removes two arcs of a single route and generates two new arcs in such a way that the section between

Published in: Networks (2017), Volume 69, Issue 3, Pages 304 - 3161
DOI: 10.1002/net.21738
Status : Postprint (Author's version)

the removed arcs is reversed. Only arc-pairs which are separated with at least four customers are considered in this neighborhood to avoid scanning the same moves as the exchange operator. The cross-exchange operator ([25]) interchanges two segments of different routes while preserving the orientation of the segments and the routes. At least one of two segments need to be of size greater than one to avoid scanning the same moves as in the exchange neighborhood. There is no upper bound on the size of the segments. Finally, the relocate operator ([28]) removes a node from its route and reinserts it in another place in its original route or in another route. This move may reduce or increase the number of routes in the solution, as relocation to an empty route is also considered. For each neighborhood, all possible moves are identified after which a best improvement strategy is applied. An overview of the local search procedure may be found in Algorithm 2.

The sequence of the neighborhoods is fixed. As relocate is also used in the perturbation phase, this operator is placed last in the local search to prevent that changes made during perturbation can be easily undone. When a local optimum is reached in a neighborhood, the local search proceeds to the next neighborhood. When a local optimum is reached in the last neighborhood, the local search procedure is repeated until no more improvement is found in any of the neighborhoods.

3.3. PERTURBATION

In the perturbation phase, customer relocation is considered once for each customer, using a randomized objective function. The general framework of the perturbation procedure may be found in Algorithm 3.

Customers are considered in random order. For each customer, a first improvement strategy is used because the goal of the perturbation phase is merely to change the current solution. It is, therefore, not necessary to choose the move with the largest improvement. The insertion positions of a customer are considered in random order, that is, the first route that is considered for insertion is chosen randomly and in each route, the first position that is considered is also chosen randomly.

The effect of relocating a customer to another position is randomized by adding a noise factor to the insertion cost. Analogous to Røpke and Pisinger [24], this noise is calculated as a random number in the interval $[- \eta * \max D, \eta * \max D]$ where η is a parameter to control the amount of noise and max*D* is the maximum distance between two nodes in the network. When the randomized insertion cost is negative or zero, the next insertion position for the customer is considered. When the randomized insertion cost is positive, the move is immediately implemented and the perturbation process continues with the next customer.

If the perturbation does not change the solution *s*, η is increased with η_{incr} and the perturbation is repeated. After *8* consecutive non-improving iterations of the ILS, a heavy perturbation is applied. This means that *n* increases with η_{heavy} to increase the level of diversification.

Algorithm 2 Local search

```
Neighborhoods = {exchange, 2-opt, cross-exchange, relocate}
s = initial solution
s' := s
stop := 0
repeat
    for i := 1 to 4 do
        next_neighborhood = 0
        repeat
            s'' ← Local search with Neighborhoods[i] on s'
            if s' = s'' then
                next_neighborhood := 1
            else
                s' := s''
            end if
        until next_neighborhood = 1
    end for
    if s = s' then
        stop := 1
    else
        s := s'
    end if
until stop = 1
```

Algorithm 3 Perturbation

```
if non_improving_it > δ then
    η := η^0 + η_heavy
else
    η := η^0
end if
repeat
    s' ← Relocate for each customer on s with noise η
    η := η + η_incr
until s ≠ s'
s := s'
```

Published in: Networks (2017), Volume 69, Issue 3, Pages 304 - 3161
DOI: 10.1002/net.21738
Status : Postprint (Author's version)

TABLE 1. Parameter list

Name	Range	Tuned value
η^0	(0.01, 0.80)	0.33
η_{heavy}	(0.0, 1.0)	0.14
η_{incr}	(0.0, 0.50)	0.22
δ	(1, 250)	196
β	(0.0, 0.50)	0.10

3.4. Acceptance Criterion

An acceptance criterion based on record-to-record travel ([9]) is applied. An overview of the procedure may be found in Algorithm 4. The solution *s* obtained after local search is always accepted to become the new incumbent solution *s* of the next ILS iteration if the cost is lower than the cost of the current best solution s^b. When the cost of *s* is higher than the cost of s^b and no heavy perturbation will be applied in the next iteration of the ILS (non_improving_it < δ), the solution is still accepted if the worsening is smaller than a certain threshold value. This threshold value is a fraction β of the cost of s^b. In case a heavy perturbation will be applied in the next ILS iteration, a worsening is never accepted in order not to deviate too far from the current best solution. In case *s* is not accepted to become the new incumbent solution, the search continues from s^b.

Algorithm 4 Acceptance criterion

```
if cost[s] < cost[s^b] then
    s^b := s
    non_improving_it := 0
else
    non_improving_it := non_improving_it + 1
    if (cost[s] > cost[s^b] · (1 + β)) or
      (non_improving_it > δ) then
       s := s^b
    end if
end if
```

3.5. Parameter Setting

To tune the parameters of our algorithm (η^0, η_{incr}, η_{heavy}, δ, β), we used the irace package, provided by Lopez-Ibanez et al. [15]. The irace package is designed for automatic algorithm configuration and implements the iterated racing procedure, which is an extension of the Iterated F-race procedure [15]. The automatic configuration process is stopped after 5000 runs of the ILS. The parameters tuned in our algorithm are given in Table 1, along with the range and tuned value.

We generated 20 test instances with sizes ranging from 20 to 75 nodes for the parameter tuning. These test instances are different from the ones used in the computational experiments in section 4.

TABLE 2. Problem classes based on parameters Q_i and L_i

	Heavy pallets ($1000 \leq \frac{Q_i}{L_i} \leq 1500$)	Mix between light ($100 \leq \frac{Q_i}{L_i} \leq 500$) and heavy ($1000 \leq \frac{Q_i}{L_i} \leq 1500$) pallets
Low variation $L_i (4 \leq L_i \leq 7)$	Problem class 1	Problem class 3
High variation $L_i (1 \leq L_i \leq 15)$	Problem class 2	Problem class 4

The number of consecutive non-improving iterations, α, after which the ILS is stopped is set to 250. This value was determined based on the results of a single run of the test instances, in which no substantial improvement was found after more than 220 consecutive non-improving iterations.

4. COMPUTATIONAL EXPERIMENTS

In this section, computational tests of the ILS on the CVRP with sequence-based pallet loading and axle weight constraints are described. All tests are performed on a Xeon E5-2680v3 CPU at 2.5 GHz with 64 GB of RAM. Different problem classes are constructed to demonstrate the performance of the model under various problem characteristics. The results are compared to those of the CVRP without axle weight constraints.

4.1. TEST SETTING

To test differences between the two models (CVRP with and without axle weight constraints), four different problem classes are created by varying the values for the number of pallets of each customer (L_i) and the total mass of the pal-lets of each customer (Q_i). In Table 2, the problem classes are presented. These problem classes are the same as used in Pollaris et al. [23]. The number of pallets may have a low variation (between 4 and 7 pallets per customer) or a high variation (between 1 and 15 pallets per customer). With respect to the weight of the pallets, axle weight constraints do not play a role when only light pallets (under 500 kg) are considered. Therefore, a distinction is made between customer demands of only heavy pallets (between 1000 and 1500 kg) and a 55% mix between customer demands with light pallets (between 100 and 500 kg) and customer demands with heavy pallets.

We use two instance sets to test the performance of the ILS. The first instance set from Pollaris et al. [23] consists of 96 instances with networks ranging from 10 to 20 customers with randomly generated coordinates for each customer between 0 and 10. The position of the depot is fixed to (5,5). This instance set is used for the validation of the ILS. The second instance set consists of 96 instances with networks ranging from 50 to 100 customers. The coordinates are generated in the same manner as in the first

Published in: Networks (2017), Volume 69, Issue 3, Pages 304 - 3161
DOI: 10.1002/net.21738
Status : Postprint (Author's version)

instance set. In all instance sets routing costs are computed by taking the Euclidean distance between the coordinates of each node pair. Values for number of pallets and total weight for each customer are generated randomly in the intervals specified in Table 2, depending on the problem class. All the instances can be found on the following website http://alpha.uhasselt.be/kris.braekers/.

An unlimited number of vehicles is considered. Characteristics of the vehicle fleet (measurements, capacity, mass, axle weight limits) are derived from information from a Belgian logistics service provider. The vehicle type that is considered is a 30-foot truck that consists of a two-axle tractor (steering axle and driving axle) and a trailer with tridem axles. In total, 22 pallets may be placed inside the truck. The total weight capacity of the truck consists of 32.2 tonnes. No more than 11.6 tonnes may be placed on the coupling, while no more than 21 tonnes may be placed on the tridem axles of the trailer. The distance from the front of the container to the coupling [parameter *c* in Equations (1) and (2)] is 1 m. The distance between the coupling and the central axle of the trailer [parameter *d* in Equations (1) and (2)] is 5.5 m. For more information regarding the vehicle characteristics, the reader is referred to Pollaris et al. [23].

4.2. Contribution of BCSP

The need for the BCSP in the generation of the initial solution is demonstrated by generating for each instance an initial solution with three different insertion methods. In the first method, an initial solution is generated by solely using a regret-2 insertion heuristic without giving special attention to *difficult* nodes. In the second method, a BCSP is solved by allowing only a single *non-difficult* node to precede each *difficult* node. The third method solves a BSCP with both a single and a combination of two *non-difficult* nodes. The regret-2 insertion heuristic was able to find a feasible solution in 132 out of 192 instances. In 187 out of 192 instances a feasible solution was found when considering only a single *non-difficult* node. For the remaining five instances a feasible initial solution was found when both a single node and a combination of two *non-difficult* nodes were allowed.

4.3. Validation of ILS

Because of the stochastic character of the metaheuristic, twenty independent runs are performed on each instance. The average and best results for each instance are reported. Results of each run on each instance can be found on the following website http://alpha.uhasselt.be/kris.braekers/. The MILP model formulated in Pollaris et al. [23] is able to solve instances up to 20 customers. To validate the performance on larger size instances, a set partitioning model is used. First, all possible routes are enumerated and checked for feasibility. Next, a set partitioning model is solved for all feasible routes in CPLEX 12.6. The computation time for the generation of the routes as well to solve the problem for the instances with 50 customers is on average 4.5 h. We were not able to find optimal solutions using this approach within a reasonable time limit for instances with 75 or 100 customers due to the large number of feasible routes.

TABLE 3. Validation ILS on CVRP without axle weight constraints

	# instances	Opt. gap Z^{best} (%)	$Z^{best} \neq Z*$	Opt. gap Z^{avg} (%)	$Z^{avg} \neq Z*$
10 customers	32	0.00	0	0.00	0
15 customers	32	0.00	0	0.06	3
20 customers	32	0.00	0	0.01	1
50 customers	27	0.04	4	0.31	26

TABLE 4. Validation ILS on CVRP with axle weight constraints

	# instances	Opt. gap Z^{best} (%)	$Z^{best} \neq Z*$	Opt. gap Z^{avg} (%)	$Z^{avg} \neq Z*$
10 customers	32	0.06	1	0.34	6
15 customers	32	0.01	1	0.35	14
20 customers	32	0.05	2	0.18	10
50 customers	27	0.07	5	1.00	27

Table 3 provides a summary of the comparison between the results of the ILS and the optimal solution for the CVRP without axle weight constraints. The ILS is able to find the optimal solution in each run for 92 out of 96 instances of the first instance set (10-20 customers). For the remaining 4 instances, the optimal solution is found in at least one run of the ILS. For the instances of size 50 of the second instance set, the set partitioning model was able to solve 27 out of 32 instances. The ILS found for 23 instances the optimal solution in at least one run. The average optimality gap of the best solution found in the ILS is 0.04%. The average optimality gap of all runs is 0.31%. For the CVRP with axle weight constraints, the optimal solutions are also found by the metaheuristic in the majority of the instances. In case the optimal solution is not found, the optimality gap is very small as well. Table 4 provides a summary of the comparison between the results of the ILS and the optimal solution. For the instances with networks of 10, 15, and 20 customers, the average optimality gaps are, respectively, 0.34%, 0.35%, and 0.18%. In 66 out of 96 instances, the optimal solution is found in all runs of the ILS. For 92 instances, the optimal solution is found in at least a single run. The set partitioning model for the CVRP with axle weight constraints is able to find an optimal solution for the same 27 instances of size 50 as the model for the CVRP without axle weight constraints. For 22 out of 27 instances the optimal solution is found in at least a single run of the ILS. The average optimality gap for all runs for the instances of size 50 is 1.00%, while the average optimality gap of the best run is only 0.07%. The results show that the ILS is able to find a set of good quality solutions for both problem types (CVRP with and without axle weight constraints).

4.4. EFFECT OF AXLE WEIGHT CONSTRAINTS

Tables 5-7 provide the results of the ILS on the instances of set 2, with networks of 50 customers, 75 customers, and 100 customers, respectively. For both models (CVRP with and without axle weight constraints), the average cost and the best cost out of 20 runs is given. For the model without axle weight constraints, the number of axle weight violations (# *V*) and maximum violation (Max *V*) (in percentage) are also reported. The number of violations represents the number of arcs traveled by a vehicle in which there is an axle weight violation. The total number of arcs traveled in which the vehicle is loaded equals the number of customers in the network. The maximum violation is expressed as a percent- age of the weight capacity of the coupling (11.6 t). In all instances, the largest violation that occurs is a violation of

Published in: Networks (2017), Volume 69, Issue 3, Pages 304 - 3161
DOI: 10.1002/net.21738
Status : Postprint (Author's version)

the weight limit on the coupling (and thus on the axles of the tractor). Violations of the weight limit on the axles of the trailer occur less frequently and are in all instances smaller than the violations on the axles of the tractor. This may be explained by the higher weight capacity of the axles of the trailer (21 t) in comparison to the weight capacity of the coupling (11.6 t). For the model with axle weight constraints, the increase in average cost compared to the average cost in the model without axle weight constraints is reported, as well as the increase in best cost compared to the best cost in the model without axle weight constraints and the average CPU time.

TABLE 5. Results of the CVRP with sequence based pallet loading with and without axle weight constraints on networks of 50 customers

Instance	Model without axle weight				Model with axle weight				
	Z^{avg}	Z^{best}	# V	Max V (%)	Z^{avg}	Z^{avg} incr (%)	Z^{best}	Z^{best} incr (%)	Avg time (s)
Problem class 1									
1	152.6	152.2	9	12	154.6	1.31	154.1	1.25	83
2	134.1	133.8	21	16	145.3	8.35	143.3	7.10	65
3	145.6	145.4	12	17	148.0	1.65	147.7	1.58	88
4	150.9	150.8	15	14	154.6	2.45	152.1	0.86	58
5	166.2	166.1	14	16	170.2	2.41	168.2	1.26	59
6	153.3	153.2	12	15	156.7	2.22	156.2	1.96	49
7	143.4	142.5	15	11	148.2	3.35	146.3	2.67	70
8	145.3	142.3	20	15	149.1	2.62	147.5	3.65	75
Problem class 2									
1	170.3	170.3	18	16	178.6	4.87	177.5	4.23	70
2	200.7	200.5	24	18	207.3	3.29	203.8	1.65	47
3	188.9	188.4	15	14	197.5	4.55	192.8	2.34	60
4	183.3	183.0	13	18	188.8	3.00	187.0	2.19	64
5	191.5	190.9	19	15	196.5	2.61	194.5	1.89	52
6	168.4	168.4	13	13	169.5	0.65	168.4	0.00	46
7	184.4	184.2	24	17	195.3	5.91	189.4	2.82	61
8	192.7	192.5	14	16	203.0	5.35	199.6	3.69	47
Problem class 3									
1	144.1	143.9	9	2	144.8	0.49	144.3	0.28	101
2	157.1	156.9	7	6	158.4	0.83	157.3	0.25	87
3	144.0	143.8	8	5	146.7	1.87	144.3	0.35	105
4	149.2	149.1	10	8	150.4	0.80	149.1	0.00	112
5	158.5	158.3	9	9	160.8	1.45	159.8	0.95	83
6	143.9	143.3	7	6	145.1	0.83	144.3	0.70	89
7	147.3	146.5	9	8	148.0	0.48	146.6	0.07	109
8	130.0	130.0	9	14	131.2	0.92	130.3	0.23	82
Problem class 4									
1	182.3	181.4	17	18	185.5	1.76	184.6	1.76	91
2	210.1	210.0	10	9	210.8	0.33	210.7	0.33	37
3	198.7	197.5	15	10	202.0	1.66	199.0	0.76	60
4	193.7	193.7	15	10	200.4	3.46	199.5	2.99	60
5	199.1	198.1	20	17	209.0	4.97	205.5	3.74	57
6	212.9	212.5	9	6	213.4	0.23	212.8	0.14	66
7	199.5	199.4	15	8	203.6	2.06	202.9	1.76	56
8	222.0	222.0	22	16	237.2	6.85	234.3	5.54	37
Average			14	12		2.61		1.84	70

V = number of violations.
Max V = maximum violation.

In the CVRP without axle weight constraints, the number of arcs in which there is an axle weight violation for networks of 50, 75, and 100 customers equals 14, 19, and 27, on average, respectively. This means that in more than 25% of the arcs that the loaded vehicles pass, there is an axle weight violation. The extent of the violations is also considerable, with on average a maximum violation of 13%, which would lead to a high fine in practice. In all instances, the solution of the ILS for the model without axle weight constraints generates axle weight violations. Results show that these violations may be avoided with a relatively small cost increase. On average the increase in average cost in the model with axle weight constraints compared to the average cost in the model without axle weight constraints is 2.61%, 2.58%, and 3.52% for the networks of, respectively, 50, 75, and 100 customers. The average increase in best cost compared to the best cost in the model without axle weight constraints is 1.84%, 1.63%, 2.39% for the networks with, respectively, 50, 75, and 100 customers. The CPU time for the instances of size 50, 75, and 100 is on average, respectively, 70, 260, 334 s.

Table 8 presents a comparison of the results per problem class and number of customers in the network. As expected, the number of violations as well as the maximum violation and the cost increase are larger in problem class 1 and 2 where only heavy pallets are considered than in problem class 3 and 4, where a 55% mix of heavy and light pallets are considered. The positive effect of mixing light pallets with heavy pallets on the costs can be explained by the fact that this allows for more flexibility in the packing process.

Published in: Networks (2017), Volume 69, Issue 3, Pages 304 - 3161
DOI: 10.1002/net.21738
Status : Postprint (Author's version)

TABLE 6. Results of the CVRP with sequence based pallet loading with and without axle weight constraints on networks of 75 customers

Instance	Model without axle weight				Model with axle weight				
	Z^{avg}	Z^{best}	# V	Max V (%)	Z^{avg}	Z^{avg} incr (%)	Z^{best}	Z^{best} incr (%)	Avg time (s)
Problem class 1									
1	201.7	201.1	20	14	207.9	3.07	203.6	1.24	210
2	208.0	206.3	24	15	216.2	3.94	213.6	3.54	214
3	208.1	207.8	25	17	215.9	3.75	213.3	2.65	213
4	213.0	212.7	17	15	219.5	3.05	217.5	2.26	218
5	200.2	199.7	17	18	205.4	2.60	201.8	1.05	204
6	200.0	198.7	21	13	208.3	4.15	202.8	2.06	206
7	213.9	213.2	22	18	223.6	4.53	219.7	3.05	224
8	207.3	206.4	23	16	215.0	3.71	212.4	2.91	216
Problem class 2									
1	273.8	273.8	28	18	288.4	5.33	283.5	3.54	286
2	260.9	260.8	21	16	266.6	2.18	262.8	0.77	267
3	300.0	298.9	32	19	312.8	4.27	308.2	3.11	318
4	275.8	275.1	25	15	285.8	3.63	282.5	2.69	286
5	323.9	323.7	26	15	332.5	2.66	329.2	1.70	335
6	263.2	262.8	19	17	267.2	1.52	264.3	0.57	267
7	315.3	314.9	28	20	327.9	4.00	323.7	2.79	331
8	326.1	325.9	31	18	334.7	2.64	333.3	2.27	334
Problem class 3									
1	221.9	220.2	10	11	225.5	1.62	223.7	1.59	224
2	195.3	194.5	11	8	198.1	1.43	196.0	0.77	197
3	201.5	201.4	15	5	206.6	2.53	203.3	0.94	209
4	204.8	203.7	11	8	209.1	2.10	204.8	0.54	211
5	205.0	204.3	14	12	207.8	1.37	205.6	0.64	208
6	229.8	229.6	12	9	232.2	1.04	230.0	0.17	232
7	202.6	202.0	12	10	204.6	0.99	202.0	0.00	203
8	211.6	209.6	11	8	215.4	1.80	212.5	1.38	218
Problem class 4									
1	337.8	337.3	24	14	342.3	1.33	340.6	0.98	341
2	326.8	326.7	22	13	335.4	2.63	331.9	1.59	333
3	328.2	327.0	14	10	334.2	1.83	332.6	1.71	334
4	325.1	322.5	18	14	330.9	1.78	326.8	1.33	332
5	271.6	270.5	13	14	274.3	0.99	271.7	0.44	274
6	287.0	286.4	17	8	290.1	1.08	287.8	0.49	292
7	288.2	287.8	22	14	297.8	3.33	295.7	2.74	296
8	269.2	268.6	11	11	273.5	1.60	270.3	0.63	277
Average			19	14		2.58		1.63	260

V = number of violations.
Max V = maximum violation.

If lighter pallets are packed first in the truck, the weight of the heavy pallets will mostly be carried by the axles of the trailer, which have a higher weight capacity. Heavy pallets are therefore better transported together with light pallets even though the total weight capacity of the vehicle is sufficient to transport solely heavy pallets. For the instances with a mix between light and heavy pallets, an increase in number of violations and maximum violation may be when we move from a low variation (problem class 3) to a high variation in number of pallets (problem class 4).

TABLE 7. Results of the CVRP with sequence based pallet loading with and without axle weight constraints on networks of 100 customers

Instance	Model without axle weight				Model with axle weight				
	Z^{avg}	Z^{best}	# V	Max V (%)	Z^{avg}	Z^{avg} incr (%)	Z^{best}	Z^{best} incr (%)	Avg time (s)
Problem class 1									
1	272.7	271.0	30	16	282.6	3.63	278.0	2.58	281
2	282.2	280.6	32	16	292.9	3.79	289.1	3.03	291
3	256.6	254.3	31	17	268.5	4.64	265.0	4.21	268
4	275.3	273.1	32	15	291.8	5.99	285.8	4.65	299
5	260.9	260.1	31	16	272.2	4.33	266.4	2.42	270
6	275.3	274.4	34	17	289.4	5.12	285.4	4.01	293
7	275.1	274.2	32	15	292.5	6.32	287.6	4.89	292
8	272.2	271.2	25	16	287.0	5.44	280.5	3.43	287
Problem class 2									
1	388.6	387.7	43	21	402.2	3.50	397.3	2.48	398
2	362.3	360.3	27	15	369.9	2.10	365.7	1.50	370
3	355.9	355.4	42	18	371.5	4.38	363.9	2.39	367
4	383.8	382.3	41	20	397.3	3.52	388.9	1.73	394
5	364.9	364.0	39	17	377.9	3.56	370.8	1.87	378
6	332.8	331.3	34	18	341.4	2.58	337.7	1.93	340
7	382.3	381.5	30	20	397.4	3.95	390.3	2.31	397
8	384.2	383.0	25	16	393.5	2.42	387.6	1.20	393
Problem class 3									
1	249.5	248.3	14	12	258.9	3.77	253.5	2.09	257
2	267.3	266.4	15	7	274.3	2.62	270.3	1.46	276
3	273.4	270.8	16	9	280.2	2.49	277.2	2.36	278
4	253.9	252.9	14	6	261.2	2.88	259.2	2.49	261
5	266.8	266.0	13	13	271.5	1.76	266.7	0.26	273
6	277.9	276.5	10	6	285.1	2.59	281.1	1.66	282
7	261.5	259.8	17	12	268.3	2.60	263.1	1.27	266
8	266.5	265.4	17	6	271.1	1.73	266.3	0.34	272
Problem class 4									
1	393.3	391.4	28	17	401.9	2.19	398.3	1.76	400
2	412.0	410.5	25	11	426.1	3.42	421.5	2.68	430
3	431.2	429.4	32	12	443.8	2.92	437.0	1.77	443
4	357.2	354.9	25	7	368.0	3.02	363.8	2.51	367
5	354.2	353.3	29	13	366.7	3.53	361.7	2.38	370
6	364.2	363.2	27	15	379.3	4.15	375.4	3.36	385
7	392.6	390.8	33	17	405.7	3.34	401.2	2.66	408
8	375.1	374.7	30	15	390.9	4.21	384.7	2.67	392
Average			27	14		3.52		2.39	334

V = number of violations.
Max V = maximum violation.

Likewise, for the instances with only heavy pallets, the number of violations and maximum violations increase when the variation in number of pallets increases. The highest number of violations and maximum violation may therefore be found in problem class 2, while the instances in problem class 3 have on average the lowest number of violations and maximum violation. An explanation for the effect of the variation on the number of violations and the maximum violation may be that a variation between 1 and 15 pallets per order leads to on average half of the orders consisting of more than 8 pallets which is much less flexible than orders between 4 and 7 pallets per customer. The probability for an axle weight violation and the extent of this violation is thereby much larger when a high variation of number of

Published in: Networks (2017), Volume 69, Issue 3, Pages 304 - 3161
DOI: 10.1002/net.21738
Status : Postprint (Author's version)

pallets is considered. An order of 15 pallets with a pallet weight of 1.4 tonnes, leads to a total weight of the order of 21 tonnes, which is less flexible to position on a truck than several smaller orders with a high pallet weight. Figure 4 provides for each problem class an overview of the variation of the average cost increase for the instances of the second instance set. Problem class 3 clearly has on average the lowest cost increase with 75% of the instances with a cost increase lower than 2.5%. In problem class 4, 75% of the instances have a cost increase lower than 3.5%. Furthermore, the variation in cost increase between the instances is much larger in problem class 4 compared to problem class 3. In problem class 1, the variance in cost increase is larger than in problem class 2, although the 25*th*, 50*th*, and 75*th* percentiles of both classes are very similar. For both problem classes, 75% of the instances have a cost increase lower than 4.6%. The variation in number of pallets per customer therefore appears to only have an impact on cost increase for the instances with a mix between light and heavy pallets (problem class 3 and 4) and not for the instances with only heavy pallets (problem class 1 and 2).

5. CONCLUSIONS AND FUTURE RESEARCH

In this article, we proposed an iterated local search method for the CVRP with sequence-based pallet loading and axle weight constraints. The ILS has proven to produce high-quality solutions, with very small optimality gaps on instances with up to 50 customers. The effect of introducing axle weight constraints in a CVRP on total routing cost is analyzed in realistic-size instances with networks consisting of 50, 75, and 100 customers. Results show that integrating axle weight constraints does not lead to a large cost increase, while not including axle weight constraints may induce major axle weight violations.

As research on vehicle routing problems with axle weight constraints is very scarce, many research opportunities still exist. Future research could integrate other realistic features in the current problem such as time windows, a heterogeneous vehicle fleet and legal driving hours. Additionally, other loading constraints may be added to the current model. Another line of possible future research could be to integrate axle weight constraints in other types of VRPs such as three-dimensional loading VRP, multicompartment VRP, and pickup and delivery problems.

TABLE 8. Summary results for each problem class

		Model without axle weight		Model with axle weight	
		# V	Max V (%)	Z^{avg} incr (%)	Z^{best} incr (%)
	50	15	15	3.04%	2.54%
Class 1	75	21	16	3.60%	2.34%
	100	31	16	4.91%	3.65%
	50	17	16	3.78%	2.35%
Class 2	75	26	17	3.28%	2.18%
	100	35	18	3.25%	1.93%
	50	9	7	0.96%	0.35%
Class 3	75	12	9	1.61%	0.75%
	100	14	9	2.55%	1.49%
	50	15	12	2.66%	2.13%
Class 4	75	18	12	1.82%	1.24%
	100	29	14	3.35%	2.47%
Average				2.90%	1.95%

V = number of violations.
Max V = maximum violation.

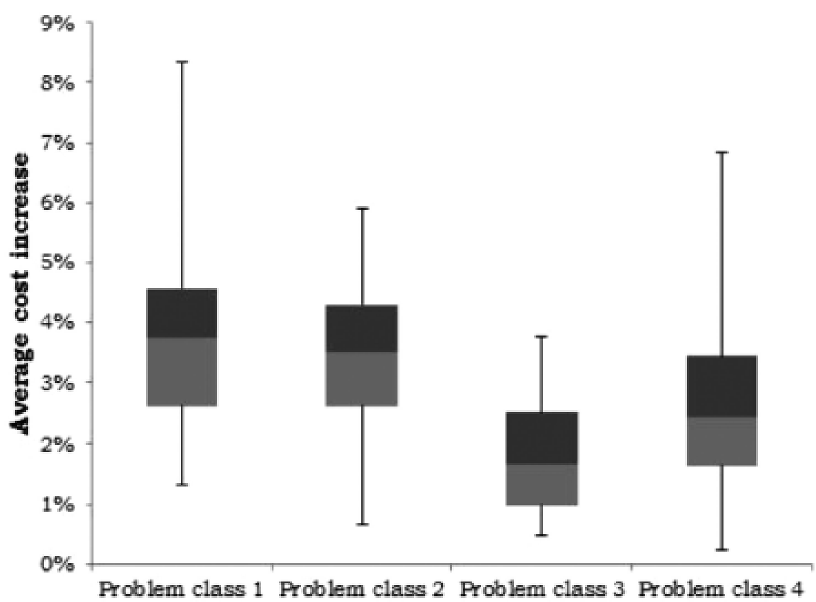


FIG. 4. Boxplot average cost increase instances set 2.

ACKNOWLEDGMENTS

The computational resources and services used in this work were provided by the VSC (Flemish Supercomputer Center), funded by the Research Foundation - Flanders (FWO) and the Flemish Government - department EWI.

Published in: Networks (2017), Volume 69, Issue 3, Pages 304 - 3161
DOI: 10.1002/net.21738
Status : Postprint (Author's version)

REFERENCES

[1] M. Alba, J. Cordeau, M. Dell'Amico, and M. Iori, A branch- and-cut algorithm for the double traveling salesman problem with multiple stacks, INFORMS J Comput 25 (2013), 41-55.

[2] M. Alonso, R. Alvarez-Valdes, M. Iori, F. Parreno, and J. Tamarit, Mathematical models for multicontainer loading problems, Omega 66 (2017), 106-117.

[3] F. Carrabs, R. Cerulli, and M.G. Speranza, A branch-and- bound algorithm for the double travelling salesman problem with two stacks, Networks 61 (2013), 58-75.

[4] P. Chen, H. Huang, and X.-Y. Dong, Iterated variable neighborhood descent algorithm for the capacitated vehicle routing problem, Expert Syst Appl 37 (2010), 1620-1627.

[5] J. Côté, C. Archetti, M. Speranza, M. Gendreau, and J. Potvin, A branch-and-cut algorithm for the pickup and delivery traveling salesman problem with multiple stacks, Networks 60 (2012), 212-226.

[6] J. Côté, M. Gendreau, and J. Potvin, Large neighborhood search for the pickup and delivery traveling salesman problem with multiple stacks, Networks 60 (2012), 19-30.

[7] G. Croes, A method for solving traveling-salesman problems, Oper Res 6 (1958), 791-812.

[8] K. Doerner, G. Fuellerer, R. Hartl, M. Gronalt, and M. Iori, Metaheuristics for the vehicle routing problem with loading constraints, Networks 49 (2007), 294-307.

[9] G. Dueck, New optimization heuristics: The great deluge algorithm and the record-to-record travel, J Comput Phys 104 (1993), 86-92.

[10] A. Felipe, M. Ortuno, and G. Tirado, Using intermediate infeasible solutions to approach vehicle routing problems with precedence and loading constraints, Eur J Oper Res 211 (2011), 66-75.

[11] A. Felipe, M.T. Ortuno, and G. Tirado, The double traveling salesman problem with multiple stacks: A variable neighborhood search approach, Comput Oper Res 36 (2009), 2983-2993.

[12] M. Iori and S. Martello, Routing problems with loading constraints, TOP 18 (2010), 4-27.

[13] B. Jacob and V. Feypell-de La Beaumelle, Improving truck safety: Potential of weigh-in-motion technology, IATSS Res 34 (2010), 9-15.

[14] A. Lim, H. Ma, C. Qiu, and W. Zhu, The single container loading problem with axle weight constraints, Int J Prod Econ 144 (2013), 358-369.

[15] M. Lopez-Ibânez, J. Dubois-Lacoste, L. Pérez Câceres, M. Birattari, and T. Stützle, The irace package: Iterated racing for automatic algorithm configuration, Oper Res Perspect 3 (2016), 43-58.

[16] H. Lourenço, O. Martin, and T. Stützle, "Iterated local search: Framework and applications," Handbook of metaheuristics, Vol. 146 of International Series in Operations Research & Management Science, M. Gendreau and J. Potvin (Editors), Springer, US, 2010, pp. 363-397.

[17] R.M. Lusby and J. Larsen, Improved exact method for the double TSP with multiple stacks, Networks 58 (2011), 290-300.

[18] R.M. Lusby, J. Larsen, M. Ehrgott, and D. Ryan, An exact method for the double TSP with multiple stacks, Int Trans Oper Res 17 (2010), 637-652.

[19] B. 0vsteb0, L.M. Hvattum, and K. Fagerholt, Routing and scheduling of roro ships with stowage constraints, Transp Res Part C Emerg Technol 19 (2011), 1225-1242.

[20] H.L. Petersen, C. Archetti, and M.G. Speranza, Exact solutions to the double travelling salesman problem with multiple stacks, Networks 56 (2010), 229-243.

[21] H.L. Petersen and O. Madsen, The double travelling sales- [25] man problem with multiple stacks formulation and heuris- tic solution approaches, Eur J Oper Res 198 (2009), 139-147.

[22] H. Pollaris, K. Braekers, A. Caris, G.K. Janssens, and S. Limbourg, Vehicle routing problems with

loading con- straints: State-of-the-art and future directions, OR Spectrum, 37 (2015), 297-330.

[23] *H. Pollaris, K. Braekers, A. Caris, G.K. Janssens, and S. Lim- bourg, Capacitated vehicle routing problem with sequence- [28] based pallet loading and axle weight constraints, EURO J Transp Logist 5 (2016), 231-255.*

[24] *S. RØpke and D. Pisinger, An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows, Transp Sci 40 (2006), 455-472.*

[25] *E. Taillard, P. Badeau, M. Gendreau, F. Guertin, and J. Potvin, A tabu search heuristic for the vehicle routing problem with soft time windows, Transp Sci 31 (1997), 170-186.*

[26] *P. Toth and D. Vigo, The vehicle routing problem, Monographs on Discrete Mathematics and Applications, Society for Industrial and Applied Mathematics, Philadelphia, 2002.*

[27] *F. Tricoire, K. Doerner, R. Hartl, and M. Iori, Heuristic and exact algorithms for the multipile vehicle routing problem, OR Spectrum 33 (2011), 931-959.*

[28] *C. Waters, A solution procedure for the vehicle-scheduling problem based on iterative route improvement, J Oper Res Soc 38 (1987), 833-839.*