CIRRELT

CIRRELT-2023-11

The Road Train Optimization Problem with Load Assignment

Eliseu Araújo
Maryam Darvish
Jacques Renaud

February 2023

Document de travail également publié par la Faculté des sciences de l'administration de l'Université Laval, sous le numéro FSA-2023-003

Bureau de Montréal

Université de Montréal
C.P. 6128, succ. Centre-Ville
Montréal (Québec) H3C 3J7
Tél : 1-514-343-7575
Télécopie : 1-514-343-7121

Bureau de Québec

Université Laval,
2325, rue de la Terrasse
Pavillon Palasis-Prince, local 2415
Québec (Québec) G1V 0A6
Tél : 1-418-656-2073
Télécopie : 1-418-656-2624

The Road Train Optimization Problem with Load Assignment

Eliseu Araújo[1], Maryam Darvish[1,2], Jacques Renaud[1,2,*]

1. Department of Operations and Decision Systems, Université Laval, Québec, Canada
2. Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT)

Abstract. This paper studies the road train optimization problem with load assignments (RTOP-LA). The RTOP-LA deals with assigning customers' demands to trailers delivered via regular trucks or road trains and determining the routing of these trucks through final customers. Road trains leave the origin terminal to reach intermediate ones, where the trailers can later be dismantled and sent to customers by regular trucks. We formulate the problem to minimize the total cost. A commercial solver is used to solve small-size instances of the problem, and we develop a multi-start iterated local search MS-ILS algorithm to obtain high-quality solutions. The results of our experiments show that MS-ILS provides optimal solutions for most instances. For small size instances MS-ILS outperforms the commercial solver, but its performance becomes more evident when the number of customers increases. Moreover, MS-ILS provides excellent solutions for larger instances in short computation times. Finally, a slightly adapted version of our algorithm has been proved efficient to solve the single truck and trailer routing problem. Compared to state-of-the-art algorithms on a set of 32 instances, our adapted algorithm obtained five new best-known solutions.

Keywords: Vehicle routing problem, road-train, multi-trailer truck

Acknowledgements. This project was partly funded by the Natural Sciences and Engineering Council of Canada (NSERC) under grants 2020-00401 and 2018-03712. We also thank our industrial partner for providing us with the data and all technical specifications. Finally, we would like to thank the associate editor and the two anonymous referees for their valuable comments and suggestions on an earlier version of this paper.

---

* Corresponding author: Jacques.Renaud@cirrelt.ca

Dépôt légal – Bibliothèque et Archives nationales du Québec
Bibliothèque et Archives Canada, 2023

1. Introduction

Greenhouse gas (GHG) emission from the road transportation sector is one of the significant sustainability challenges for cities. Road transport emissions, representing 21% of total GHG emissions in Canada, increased by 12% from 2005 to 2017, mainly due to the increase in the number of vehicles on the roads and the increased use of large, heavy trucks (Canada, 2020). In the province of Quebec, the second-largest province in Canada with 8.3 million inhabitants, this proportion goes up to 35.6% (Gouvernement du Québec, 2019). While transportation companies consistently seek ways to reduce fuel consumption and carbon footprint and improve efficiency, city officials also introduce and implement various solutions to promote sustainability. For example, in Quebec, the government has permitted the travel of road trains on highways. The road train technology aims to minimize total operating costs by consolidating transportation (Keaton, 1991) to increase efficiency and throughput.

Road trains (also referred to as multi-trailer trucks, long-combination vehicles, or turnpike double) consist of two or more (up to half a dozen) trailers or semi-trailers hauled by a tractor truck (Boysen et al., 2017). In general, long-combination vehicles (LCVs) have three major truck configurations: a tractor-semi-trailer with a trailer up to 28 feet (known as the Rocky Mountain Double), a tractor hauling two trailers up to 48 feet each (known as the Turnpike Double), a tractor hauling three trailers up to 28 feet each, known as the Triple Trailer Combination (Dessouky et al., 2007). Due to safety reasons, in most countries, the maximum allowed dimension and combination of vehicles are regulated (Jagelčák et al., 2019). For example, the use of 33-feet doubles and 28-feet triples are prohibited in some states in the US (Chen et al., 2019) or some European countries such as Denmark, the Netherlands, Sweden, Finland, and Norway. However, in some German federal states, vehicle combinations of up to 25.25 meters long is permitted (Jagelčák et al., 2019). In Canada, an LCV (road train) consists of a tractor and two trailers with a combined length of 25 to 40 meters (Lightstone et al., 2021). Under certain regulations, four Canadian provinces (Alberta, British Columbia, Ontario, and Quebec) support the use of road trains (Lightstone et al., 2021). In the province of Quebec, the regulation allows a road train to haul two trailers of up to 16.2 meters each (Ministère des Transports, 2020).

Road trains are generally routed through intermediate terminals, and the literature shows operational gains when using them (Guastaroba et al., 2016; Atefi et al., 2018). These intermediate terminals are mainly known to improve distribution logistics since each customer's demand can be assigned to and shipped from the closest terminal in a more decentralized manner. Therefore, the main gain comes from resource-saving offered by the economies of scales (Crainic and Kim, 2007). Coupling two or more trailers into a truck to travel to these intermediate terminals brings an even more considerable operational advantage. It saves directly one truck, one driver, and the corresponding fuel and wages. Intermediate terminals allow sets of demands to be brought to locations close to final customers and from them to be routed to final destinations. However, if customers are located relatively far from these terminals, several smaller vehicles are still needed for final deliveries. Therefore, how to optimize the routes to move trailers through the distribution network using a mix of road trains and regular trucks requires a rigorous approach.

Inspired by a real-world Quebec-based transportation company, we describe and model a problem, which will be called the *road train optimization problem with load assignment* (RTOP–LA). We consider two types of orders. The first type either requires a complete trailer or is large enough to be delivered directly as a *full truckload (TL)*. Other orders are smaller and will be consolidated to be delivered via *less-than-truckload* (LTL) deliveries.

The mechanical configuration of trucks allows for moving one or two trailers at a time. In the following, we refer to these configurations as *regular trucks* versus *road trains*, respectively. Road trains can move several trailers simultaneously with either TL or LTL orders. Due to the existing legislative rules and regulations in Quebec, road trains can only travel between terminals on highways and cannot deliver directly to final customers. It should also be noted that setting up a road train is a complicated task, as the attachment of the second trailer requires delicate operations. Therefore, only some drivers can operate a road train. When road trains arrive at the destination terminal, they are dismantled, and then each order is delivered to the final destination by a regular truck. Now given a fleet of trucks, the goal is to minimize the transportation cost and, consequently, the fuel consumption of all these deliveries. For computational purposes and inspired by this real case, we generate several problem instances varying in size and features, such as the demand patterns. We formulate the RTOP–LA as a mixed-integer linear programming (MIP) problem. A commercial MIP solver could optimally solve several small instances of the problem. However, to solve real-size instances of the problem, we have developed a Multi-Start Iterated Local Search (MS-ILS) algorithm.

In summary, the contributions of this paper are as follows: i) we formally introduce the RTOP–LA and propose a mathematical model for it, ii) we develop an MS-ILS algorithm to obtain solutions for real-size instances of the problem, for which a commercial solver proved to be limited, iii) we also validate our algorithm by solving available benchmark instances of the single truck and trailer problem from the literature, and finally, iv) we perform a sensitivity analysis on the number of trailers loaded on a road train.

The remainder of this paper is organized as follows. Section 2 reviews similar problems from the literature. In Section 3, we describe the problem, followed by the mathematical formulation presented in Section 4. A numerical example is given next in Section 5. The proposed solution algorithm is elaborated in Section 6. We present the results of the computational experiments in Section 7, followed by conclusions in Section 8.

2. Literature review

In this section, we review the relevant literature on road train optimization. Although still scarce, road trains are mentioned in the optimization literature. Laumanns et al. (2001) demonstrated the advantages of using road trains in reducing transportation costs compared to using other types of vehicles. They developed a multi-objective mathematical model to optimize fuel consumption and speed jointly. The authors proposed evolutionary algorithms and showed the Pareto points closest to optimal Pareto solutions. From the computational perspective, although the method could obtain solutions close to the set of optimal Pareto ones, the authors attested to difficulties in getting good results. Dessouky et al. (2007) conducted a simulation study to present the efficiency of using shorter and lighter multiple trailers instead of one long and heavy trailer.

The RTOP–LA is related to three vehicle routing problem (VRP) streams: i) Truck and trailer routing problem (TTRP), ii) Swap-Body VRP, and iii) two-echelon VRP. In what follows, we briefly discuss recent contributions in each of these streams and show their relevance to the RTOP–LA.

It should be noted that ideas such as the use of truck platoons (Bhoopalam et al., 2018) or doubly open park-and-loop routing problem (Cabrera et al., 2022) are also loosely related to the RTOP–LA. We refer the reader to the excellent work published recently (see Boysen et al. (2021)).

2.1. Truck and trailer routing problem

The RTOP–LA shares similarities with the TTRP which was first introduced as an extension of the VRP in Semet (1995) where two sets of customers, vehicle and truck customers, are defined. The vehicle customers can be visited by a truck coupled with a trailer, whereas the truck customers do not require the trailer. Although several different costs, such as parking the trailer, loading and unloading (Chao, 2002) can occur in the TTRP context, mainly the objective is to minimize the total travel costs. Chao (2002) proposed a tabu search (TS) algorithm to solve benchmark instances. Lin et al. (2009) applied a simulated annealing heuristic to the TTRP which could improve the solutions of several benchmark instances.

Villegas et al. (2010) presented a problem called the single truck and trailer routing problem with satellite depots (STTRPSD). In this problem, a truck with a trailer visits a set of satellite depots. At a satellite depot, the trailer is detached, and the truck visits the customers assigned to this satellite depot. Similar to our problem, a truck with a trailer can only travel on main roads. The authors solved the problem in two levels; in the first one, the truck with the trailer departs from the main depot to visit a subset of trailer points. In the second level, customers are visited by the truck alone. Derigs et al. (2013) introduced the idea of load transfer between truck and trailer and imposing time windows for delivery. They developed a local search based heuristic. They concluded that the definition of the problem is rather simplistic and the benchmark instances from the literature do not show the full complexity of the problem. Therefore, they called for more realistic problem settings and benchmark instances. Accorsi and Vigo (2020) introduced a general variant of the single truck and trailer routing problem and proposed a unified solution approach for a class of problems.

Drexl (2013) surveyed the applications of the vehicle routing problem with trailers and transshipments (VRPTT). In the VRPTT, the main questions to answer are *"which vehicle transfers how much load, when, where and into which other vehicle?"*

2.2. Swap-Body VRP

The Swap-Body VRP (SB-VRP) is a generalization of the TTRP (Absi et al., 2017). A swap body is a type of freight container used for road and rail transport. These containers are standardised. Folding legs under their frame is their main characteristic. A homogeneous fleet consisting of trucks, semi-trailers, and swap-bodies as well as a set of swap locations are considered in the SB-VRP (Todosijević et al., 2017; Miranda-Bront et al., 2017). The swap locations are similar to the intermediate terminals in the RTOP–LA where the swap-bodies (trailers) can be exchanged (Miranda-Bront et al., 2017). The vehicles can pull up to two swap-bodies (Absi et al., 2017). A truck carrying two SBs is called a train. Some customers can be visited by a train, some others only by a truck, carrying one SB, and some are mandatory train customers (Huber and Geiger, 2017). The goal of the SB-VRP is to minimize the total cost, including the fixed costs of using the vehicles and the operational costs of performing the routes (Todosijević et al., 2017).

Absi et al. (2017) considered a relax and repair heuristic. Todosijević et al. (2017) presented a mixed integer programming formulation for the SB-VRP and proposed two general variable neighborhood search heuristics to solve the benchmark instances. In Huber and Geiger (2017), the performance of several neighborhood operators to solve the SB-VRP is studied. They also investigated the use of parallel variable neighborhood searches to solve the problem where the master thread collects the best solutions obtained on different parallel threads. Miranda-Bront et al. (2017) developed a cluster-first route-second heuristic with greedy randomized adaptive search and iterated local search procedure metaheuristics to solve it.

RTOP–LA is similar to the SB-VRP since, in both problems, trucks or trains leave from a depot to several intermediate/swap locations. In the RTOP–LA, road trains are dismantled at the terminal and only trucks are allowed to visit customers whereas in the SB-VRP, depending on the type of customer, a truck or a train visits the customer. Moreover, in the SB-VRP, a truck/train must have the same swap bodies with which it has departed when back to the main depot (Huber and Geiger, 2017).

2.3. Two-echelon VRP

Finally, RTOP–LA shares similarities with the two-echelon VRP since trucks/road-trains are dispatched from a main terminal to visit several intermediate terminals before the freight is moved to the final customers. Several surveys exist in the literature on the two-echelon VRPs (see, Perboli et al. (2011); Cuda et al. (2015); Sluijk et al. (2023)). The use of intermediate facilities (e.g., cross-docks, distribution centers, and terminals) in multi-echelon distribution networks is very well studied in the optimization literature. Studies on the multi-echelon routing problems are also numerous (e.g., Dondo et al. (2011); Baldacci et al. (2013); Darvish et al. (2019); Smith et al. (2021)). See Cuda et al. (2015) who provided an overview of the two-echelon routing problems.

Previous studies show how using terminals may help route optimization, which in turn contributes to fuel consumption minimization and other related operating expenses. Our literature review shows that although multi-echelon VRP, SB-VRP, and TTRP are problems broadly studied in the literature, distribution problems considering the use of road trains with intermediate terminals, especially considering their real applications are still very limited.

3. Problem description

We define the RTOP–LA on an undirected graph $G = (N, E)$ where $N = \{0, 1, \ldots, m+n\}$ is the set of all nodes and $E = \{(i, j) : i, j \in N\}$ is the edge set, where $i < j$. The node set N is partitioned into three sets, $N = \{0\} \cup T \cup J$; $T = \{1, 2, ..., m\}$ is the set of terminals where 1 is the origin terminal and the $m - 1$ others are the intermediate ones, also called satellite terminals, and $J = \{m + 1, ..., m + n\}$ is the set of final customers.

Node 0 is the dummy node used for modelling routing variables. In this regard, we also define $V = \{0\} \cup J$.

Each customer $j \in J$ has a demand q_j, representing the number of pallets ordered (as an integer value). A trailer can hold up to Q pallets, therefore, if a customer order is equal to the trailer capacity, $q_i = Q$, this customer is said to have ordered a TL, and when it is less than the trailer capacity, typically less than 75%, it is said to have ordered an LTL. Trailers can be loaded either on regular trucks or on road trains. In our context, the company owns several trucks and 53-foot trailers that can be hauled alone or as a pair on a road train. Therefore, there is no limit associated with the fleet. A truck can move either one or two trailers, but a regular truck driver cannot drive a road train. Therefore, the number of available road trains is limited by the number of available experienced drivers r, and we assume no limit on the number of available regular truck drivers.

Requests are known at least a day in advance; thus, we assume that demands are deterministic. A TL order is delivered to a single customer at a time by a back-and-forth route. For customers having LTL orders, we need to determine the routes associated with each terminal.

In the following, we consider that a regular truck, which moves only one trailer at a time, has a unit cost of c_1 per kilometer. These trucks (and drivers) are available at the origin terminal and every intermediate terminal (those of set T), and they can visit any node of J. Using a road train incurs a unit cost of c_2 per kilometer. Typically c_2 is $25-40\%$ higher than c_1 as setting up a road train requires delicate operations and has to be carried out by experienced drivers. If we define d_{ij} as the distance in kilometer between nodes i and j, then $c_{ij} = c_1 \times d_{ij}$ represents the cost associated with the edge (i, j) for regular trucks and $\bar{c}_t = c_2 \times d_{1,t}$ with $t \in T \backslash \{1\}$ the cost of sending a road train to an intermediate terminal t. Therefore, c_{ij} represents the cost associated with the edge (i, j) for regular trucks and $\bar{c}_t$ the cost of sending a road train to an intermediate terminal t.

A road train travels through highways from the origin terminal to another intermediate terminal. Once at the destination terminal, the road train is dismantled, and the two trailers (TL or LTL) are delivered to their final destinations by regular trucks. To solve the problem, one needs to determine the followings:

(i) the orders assigned to each trailer,

(ii) the number of trailers (TL or LTL) assigned to regular trucks departing from the origin terminal to deliver directly to final customers,

(iii) the number of trailers (TL or LTL) assigned to road trains departing from the origin terminal to the intermediate ones,

(iv) for each intermediate terminal that receives a road train, eventually, one needs to:

- assign trailers' load to regular trucks to deliver to final customers and,
- make sure that each regular truck returns to its terminal (the one from which it had departed).

It should be noted that the loading decisions are handled by assigning orders to trailers and trailers to trucks.

4. Mathematical formulations

In this section, we present the mathematical formulation for the RTOP–LA. The following variables are defined. The assignment of trailer k to terminal t is denoted by y_{tk}. z_{tkj} equals one if customer j is served from terminal t by a truck carrying trailer k and w_t denotes the number of road trains sent to terminal t. Finally, the number of times edge (i, j) is traversed by a truck carrying trailer k from terminal t is denoted by x_{ijtk}. x_{ijtk} are the routing variables, and they are defined from each terminal. Therefore $i \in J \cup \{0\}$, where node 0 is the dummy starting node for each route. For example, if $x_{0j2k} = 1$, it means that customer j is visited from terminal 2 by a truck carrying trailer k. Table 1 summarizes the notation of our model.

Table 1: Parameters, sets and variables of model RTOP–LA

Parameters	
i, j	node index
t	terminal index
c_{ij}	cost associated with the edge (i, j), for regular trucks
$\bar{c}_t$	cost of sending a road train to terminal t
q_j	demand of customer $j \in J$
Q	trailer capacity
r	number of road train drivers
Sets	
T	set of terminal sites
J	set of customers
V	set of customers and a dummy terminal ($\{0\} \cup J$)
K	set of trailers
E	set of edges
S	all subsets of J such that $\sum_{j \in S} q_j \leq Q$
Variables	
y_{tk}	if trailer k is dispatched from terminal t,
x_{ijtk}	the number of times edge (i, j) is traversed by a truck carrying trailer k from terminal t where $i \in J \cup \{0\}$ and $j \in J$, $i < j$,
w_t	number of road trains sent to terminal t,
z_{tkj}	if customer j is visited by trailer k from terminal t.

$$Min \sum_{i \in V} \sum_{\substack{j \in J \\ i<j}} \sum_{k \in K} \sum_{t \in T} c_{ij} x_{ijtk} + \sum_{t \in T \setminus \{1\}} \bar{c}_t w_t \tag{1}$$

Subject to

$$\sum_{t \in T \setminus \{1\}} w_t \leq r \tag{2}$$

$$\sum_{k \in K} y_{tk} = 2w_t \quad \forall t \in T \setminus \{1\} \tag{3}$$

$$\sum_{t \in T} y_{tk} \leq 1 \quad \forall k \in K \tag{4}$$

$$\sum_{j \in J} q_j z_{tkj} \leq Q y_{tk} \quad \forall k \in K \quad \forall t \in T \tag{5}$$

$$\sum_{t \in T} \sum_{k \in K} z_{tkj} = 1 \quad j \in J \tag{6}$$

$$\sum_{\substack{i \in V \\ i<j}} x_{ijtk} + \sum_{\substack{i \in V \\ i>j}} x_{jitk} = 2z_{tkj} \quad \forall j \in J \quad \forall t \in T \quad \forall k \in K \tag{7}$$

$$\sum_{j \in J} x_{0jtk} = 2y_{tk} \quad \forall k \in K \quad \forall t \in T \tag{8}$$

$$\sum_{i \in S} \sum_{\substack{j \in S \\ i<j}} x_{ijkt} \leq |S| - 1, \forall S \subseteq J \quad k \in K \quad \forall t \in T \tag{9}$$

$$x_{ijtk} \in \{0,1,2\} \quad \forall t \in T \quad k \in K \quad i \in V j \in J \tag{10}$$

$$y_{tk} \in \{0,1\}, \forall t \in T \quad k \in K \tag{11}$$

$$z_{tkj} \in \{0,1\} \quad \forall t \in T \quad k \in K \quad j \in J. \tag{12}$$

$$w_t \in \mathbb{Z}^* \quad \forall t \in T \tag{13}$$

The objective function (1) minimizes the total transportation cost of regular trucks and road trains. The cost includes the routing costs and the cost of sending road trains from the original terminal to the intermediate ones. Constraint (2) limits the number of trailers sent by road trains to intermediate terminals, which is twice the number of available road train drivers. Constraints (3) link the total number of trailers assigned to each terminal with the total number of road trains sent to the same terminal. Constraints (4) assure that a trailer is assigned to at most one terminal. Constraints (5) are the trailer capacity constraints. Constraints (6) assure that each customer must receive a trailer from a terminal. Constraints (7) and (8) are the degree constraints. Constraints (9) are the subtour elimination constraints. These constraints are used as lazy constraints which means that they are added to a pool of cuts that are not initially active. As soon as a feasible solution is found, violated subtour constraints are detected. It should be noted that when the set of LTL orders is non-empty, the RTOP–LA is $\mathcal{NP}$-hard since the routing problem at each terminal reduces to a vehicle routing problem or a travelling salesman problem (see Laporte and Nobert (1987) and Laporte (1992)). More specifically, RTOP–LA can be reduced to a two-echelon capacitated VRP, which for the case with even one satellite depot is proved to be $\mathcal{NP}$-hard (Perboli et al., 2011). This explains the presence of constraints (9). Constraints (10)–(12) detail the domain of the variables used in the model.

5. Numerical example

In this section, we provide a numerical example to clarify the problem encountered by our industrial partner. In this example, we consider twenty customers, three terminals, and five road train drivers. Terminal 1 is the origin terminal and Terminals 2 and 3 are the intermediate ones. Trailers are assumed to have a capacity of 24 pallets.

Figure 1 shows a solution with three road trains (dashed lines) and nine regular trucks. The solution is presented on the right hand side of the figure, where we see five LTL load assignments to trailers, respecting the capacity $Q = 24$. We also see TL order assignments. On the left hand side, red lines show routes with multiple customers, and green lines are single-customer routes. Customers are shown by red ovals and the terminals by green rectangles. The first road train brings two TL trailers to terminal 2 and later from this terminal, they are delivered to customers 4 and 5 by two regular trucks. The second road train moves two LTL trailers to terminal 2. Then two regular trucks perform routes to customers 6 and 18 and to customers 11, 9, 16, and 8. The third road train carries two LTL trailers to terminal 3 where a regular truck meets the demands of customers 7, 15, and 19, and a second one fulfils the demands of customers 12, 13, 2, 20, and 14. Finally, customers 1 and 17 receive their TL demands from the main terminal and customers 3 and 10 are served in LTL from the main terminal.

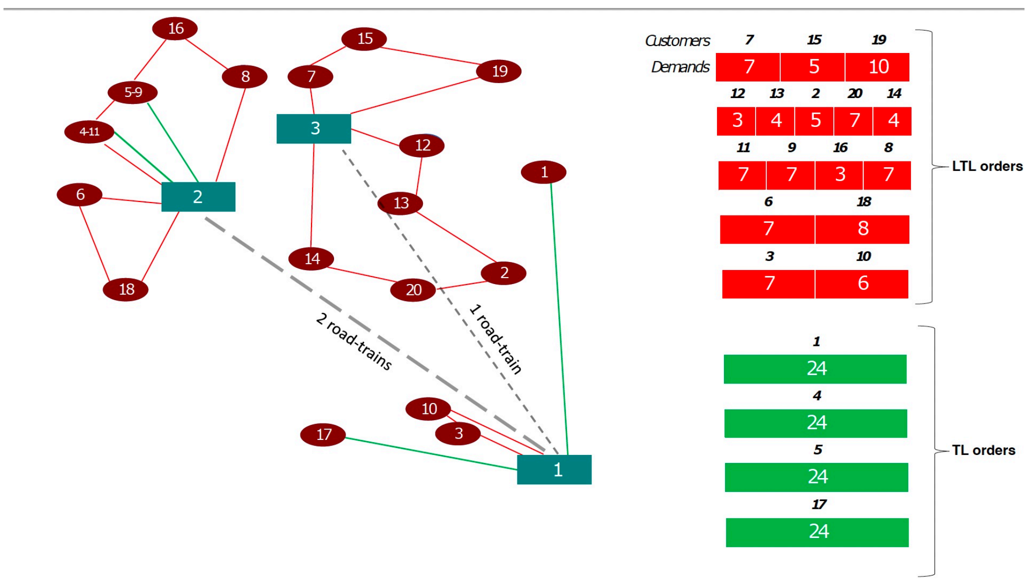


Figure 1: An example of the RTOP–LA solution

6. Solution algorithm

To date, many intricate metaheuristics are proposed to solve two-echelon VRPs. Among them, the ILS has shown promising results in dealing with routing and location-routing problems (Penna et al., 2013). The ILS (Lourenço et al., 2019) is a metaheuristic that applies local searches to solutions generated through perturbations. These perturbations are neighborhood structures that alter solutions, aiming to increase diversity and expand the search space. After a perturbation process, a local search is generally applied, defining a region for the search space with the goal of intensification. Moreover, Iterated Tabu Search (ITS), which combines ILS with a TS, has been effective in solving several combinatorial optimization problems (Lai and Lü, 2013). Given the success of the ILS in solving similar problems, we have developed a multi-start ILS (MS-ILS) with tabu list metaheuristic to solve the RTOP–LA. This algorithm has several features as follows.

As suggested in Nguyen et al. (2012), we use a multi-start framework to diversify the solutions without wasting time in unproductive iterations. $restart_{max}$ is the number of times the MS-ILS is restarted from an initial solution.

The second feature is the local search component which tries to improve the current solution obtained from the perturbation procedure. Moreover, a TS list is introduced to prevent revisiting the local optimums.

The third feature is a perturbation operator. The goal is to avoid getting stranded in a local optimum. The number of perturbations is set to p_{max}.

The general process is summarized in Algorithm 1. In what follows, we elaborate the components of the proposed MS-ILS.

Algorithm 1: Multi-start iterated local search.

```
Result: s*
Iter = 1;
repeat
    s0 ← RandomSolution() ;                /* Initialization: Algorithm 2 */
    s ← LocalSearch(s0) ;                  /* Local search: Algorithm 3 */
    p_iter = 0
    while p_iter ≤ p_max ;                 /* Perturbation loop */
     do
        s' ← Perturb(s) ;                  /* Perturbation: Algorithm 4 */
        s'' ← LocalSearch(s') ;            /* Local search: Algorithm 3 */
        if s'' is feasible then
            s ← s'' ;
        if f(s) < f(s*) then
            s* ← s;
            p_iter = 0 ;
        else
            s ← s*;
            p_iter = p_iter + 1;
    Iter = Iter + 1;
until Iter = restart_max;
```

6.1. A random initial solution

A random initial solution is generated heuristically. We start by constructing the regular truck routes followed by road train ones. A customer, a terminal, and a truck are randomly selected. The customer is added to a route (selected truck) linked to the selected terminal. This customer is removed from the set of customers to be visited. From the set of unvisited customers, another customer is selected and if feasible, with respect to the capacity limit of the truck, the customer is added to the route. The random selection of customers, the feasibility check, and their insertion into the route continue until the truck reaches its capacity limit. After each selection, the customer is inserted in the lowest cost position.

The maximum number of available trucks at each terminal is set to twice the number of road train drivers, r. As many road trains as possible are formed by pairs of truck routes coming from the same terminal. The procedure is summarized in Algorithm 2.

Algorithm 2: RandomSolution().

Data: J, K, T
Result: s_0
repeat
  Select randomly trailer $k \in K$, terminal $t \in T$;
  Set $\overline{J} = \emptyset$;
  repeat
    Select randomly customer $j \in J$
    if *feasible with respect to trailer capacity* **then**
      Find the cheapest insertion position for customer j in the route of trailer k to terminal t;
      Remove j from J;
    else
      Move customer j from J to $\overline{J}$;
  until $J \neq \emptyset$;
  Remove k from K
  Remove t from T if its number of associated routes reaches r
  Set $J = \overline{J}$
until $J \neq \emptyset$;
Form road trains by pairing truck routes from the same terminal;

6.2. Local search algorithm

We use a strategy to combine a variable neighborhood descent (VND) method (Hansen and Mladenović, 2014) with a tabu search algorithm. A set of local search operators (LS) is available, and one of them (operator l) is randomly selected at the beginning of each iteration. This procedure is known as the VND with a random neighborhood ordering (RVND) successfully applied to several fleet vehicle routing problems (see, Subramanian et al. (2010); Penna et al. (2013)).

Our preliminary tests revealed that the first-improve strategy works better than the best-improve one. Therefore, we apply the first-improve strategy to the local search operator. If the new solution does not improve the incumbent solution, then l is removed from LS, a different local search operator is selected, and the process restarts. All local search operators are returned to LS set if the solution is improved. The process continues until LS is empty. The current solution is maintained if the process does not find an improved solution. During this procedure, whenever a new solution is found, its value and the details of the routes are saved in a *tabu list*. The goal is to ensure the diversity of solutions and to avoid local optima as local searches cannot use the tabu listed solutions. The length of the tabu list is between $tabu_{min}$ and $tabu_{max}$. Each time a new solution is found, the list length is reduced to $tabu_{min}$. The local search followed by tabu list updating procedure ends after $Iter_{max}$ iterations with no improvements.

In this paper, we use three classical local searches as proposed in Subramanyam et al. (2020). The *relocate* procedure takes one customer at a time and tries to insert it in all other positions and in all routes. The *exchange* procedure takes all pairs of customers from the same route and exchanges their positions. Finally, the *inter-route 2-opt* chooses two routes from the same terminal and deletes an arc from each of them. Then, all possible links between nodes of the removed arcs are tested. The best option that reconstructs the route is considered as the new current solution. The procedure is repeated for all pairs of arcs. The pseudocode of the local search algorithm is provided in Algorithm 3.

6.3. Neighborhood structure for the perturbation

At each perturbation iteration, the process starts by selecting two routes from the set of routes, R. The first route is the one having the highest average demand for visited customers. The second one is selected by looking at the distances between the customers of this route and all other customers. We identify the pair of customers having the shortest distance. The route that contains this other closest customer is called *closest route*. These two routes, i.e., the one with the highest average demand and the closest route, are then removed, and all their customers are placed in V', the set of unvisited customers, and then relocated randomly to other routes (trucks) with available capacities. If the total capacity of the trucks does not allow the insertion of these customers, new trucks are allocated

Algorithm 3: LocalSearch().

```
Data:  s, tabu_min, tabu_max, Iter_max
s* ← s;
Iteration = 0;
current_size = tabu_min;
repeat
    Randomly select a local search operator l ∈ LS;
    repeat
        Iteration = Iteration + 1;
        s' = first improvement l(s);
        if f(s') < f(s) then
            s ← s';
            if |tabu list| > current_size then
                Remove the first position solution from tabu list;
            end
            if f(s') < f(s*) then
                s* ← s';
                Iteration = 0;
                current_size = tabu_min;
            end
            Add s' to tabu list;
            Reset LS with all local searches;
            Select a local search operator l ∈ LS;
        else
            Remove l from LS;
            Select l' ∈ LS;
            l ← l';
            if current_size < tabu_max then
                current_size = current_size + 1;
            end
        end
    until  LS = ∅;
    Reset LS with all local searches;
until Iteration = Iter_max;
```

Algorithm 4: Perturb().

```
Data:  s, R
Result: s'
From R select route r with the highest average demand;
From R select route r' as the closest route of r;
Remove all customers from r and r' and place them in V';
repeat
    Select a customer from V' and a random route from R;
    if feasible with respect to trailer capacity then
        Relocate the customer randomly in the selected route;
    else
        Add a route to R originating from the main terminal;
        Add the selected customer to this route;
    end
    Remove the customer from V';
until V' = ∅;
```

to their original terminals and customers are inserted in these routes. Next, we apply a random local search operator on this perturbed solution, as presented in Algorithm 1. The road train set is updated next by forming pairs of routes that leave from the same terminal. This process is repeated p_{max} times. Algorithm 4 summarizes this procedure.

7. Computational results

This section presents and discusses the results of our computational experiments. The model presented in Section 4 and the proposed solution algorithm elaborated in Section 6 are coded in C++. The model is solved using Gurobi Optimization solver 9.5.0. All computational experiments are conducted on an Intel Core i7 processor running at 4.0 GHz with 64 GB of RAM installed with the Ubuntu Linux operating system.

In this section, first, we start by introducing the instances generated in this paper in Section 7.1. Then in Section 7.2, the performance of the model in solving these instances is evaluated. The parameter tuning procedure for the proposed algorithm is described in Section 7.3. Its performance is evaluated in Section 7.4 and by solving instances for the single truck and trailer with intermediate depot in Section 7.5. Finally, in Section 7.6, we perform a sensitivity analysis on the road train configuration concerning the number of trailers loaded.

7.1. Instance generation

Inspired by our collaboration with a major common carrier, we generated several instances with different combinations of orders. An order is classified as a truck-load (TL) if it corresponds to a trailer's capacity, thus 24 pallets. For less-than-truckload (LTL) orders, we used different demand patterns as presented in Table 2.

Table 2: Instance set parameters

Number of instance sets	Number of TL	Number of LTL	LTL Demand Pattern	
			High	Low
40	4, 8, 12, 16	4, 8, 12, 16, 20	[7, 8, 9]	[4, 5, 6]

We use four values for the number of TL and five for the number of LTL orders and define two demand patterns for the LTL orders. When the number of pallets is 7, 8, or 9, the demand pattern is considered as high and when it is either 4, 5 or 6, the demand is considered low. As the trailer capacity, Q, is set to 24 pallets; high demand will result in routes with, on average, three deliveries and between four to six deliveries for the low demand pattern. We generate 40 instance sets and four instances in each set for a total of 160 individual instances.

The coordinates of each customer are integer values randomly placed in a 50×50 area. We place the main terminal within $x \in [30, 50]$ and $y \in [0, 15]$ (lower right corner). The coordinates of the first intermediate terminal are in $x \in [10, 35]$ and $y \in [20, 30]$ and the second one in $x \in [0, 25]$ and $y \in [30, 45]$.

These instance sets are numbered as TLa–LTLb–c where a is the number of TL, b the number of LTL, and c is either H or L for the demand pattern, referring to high versus low demand pattern. Thus, instances in set TL12–LTL20–H have 12 TL orders and 20 LTL orders with a high demand pattern. It should be noted that instances with 8 TL and 16 LTL are representative of the company's day-to-day operations. The instances are publicly available at https://doi.org/10.5683/SP3/R6ZNPW.

7.2. Performance of the model

The model presented in Section 4 is solved using a commercial solver (Gurobi solver). Table 3 shows the average results obtained where the first two columns provide information on the instance. Then for high and low demand patterns, we report the solution value or the upper bound (UB), the percentage gap (Gap%) and the computing time (Time) in seconds. For instances solved to optimality, $Gap = 100 \times (UB - z^*)/z^*$ where z^* is the value of the optimal solution. When z^* is unknown, we use $Gap = 100 \times (UB - LB)/LB$ where LB is the lower bound reported by Gurobi (see Laporte and Toth (2022)). The average results are from four instances generated for each combination of TL and LTL. Note that the time limit is set to 21,600 seconds.

Table 3: Average of results obtained by the commercial solver

		High demand pattern			Low demand pattern		
TL	LTL	UB	Gap(%)	Time (s)	UB	Gap (%)	Time (s)
4	4	319.72	0.00	3	279.40	0.00	1
4	8	322.60	0.00	1136	322.25	0.00	220
4	12	455.93	3.67	17814	379.50	7.14	21600
4	16	592.92	10.84	21600	448.06	17.55	21600
4	20	539.80	16.96	21601	459.75	13.49	21600
8	4	423.03	0.00	2	404.15	0.00	2
8	8	495.00	0.00	334	506.45	0.00	491
8	12	574.21	5.08	21600	523.26	2.24	18644
8	16	598.22	6.76	21600	604.16	6.57	21600
8	20	769.31	15.35	21600	618.00	12.52	21600
12	4	566.09	0.00	8	566.21	0.00	6
12	8	642.32	0.00	756	671.33	0.00	238
12	12	786.52	3.44	21522	781.36	1.99	13145
12	16	888.30	5.60	21600	697.40	5.59	21600
12	20	952.90	13.24	21600	808.46	8.09	21600
16	4	726.64	0.00	13	769.13	0.00	14
16	8	845.72	0.00	485	744.66	0.00	727
16	12	807.45	2.10	17428	834.88	0.00	743
16	16	1080.79	10.25	21600	894.56	7.09	21600
16	20	1044.32	11.09	21601	973.88	8.63	21600
Global average		**671.59**	**5.22**	**12695**	**614.34**	**4.54**	**11432**

On average and over all instances, we observe a gap of 5.22% and 4.54% to optimal solutions for high and low demand instances, respectively. We also observe that the low demand instances are slightly easier to solve, as shown by the obtained optimality gap and the execution time. The results in Table 3 also show that all instances with four or eight LTLs can be solved to optimality on average in 353 seconds. However, as the number of LTL deliveries increases, the solver struggles to prove optimality, in most cases taking the whole three hours allocated running time.

7.3. Tuning parameters

For the MS-ILS parameter tuning, we follow the simple sequential strategy proposed by Accorsi and Vigo (2021). First, we consider a default reasonable value for each parameter to tune. These values are set based on preliminary trial-and-errors or the literature (e.g., (Nguyen et al., 2012)). Then, we change the value of one parameter keeping the rest as default to get the cost and the execution time. The goal is to identify a set of parameters leading to good quality solutions while keeping the running time low (Accorsi and Vigo, 2021).

We perform the parameter tuning on an independent set of 10 instances as the benchmark. In these instances, the number of LTL orders is fixed to 20, and we use 4, 8, 12, 16 and 20 TL orders. For each combination, we generated an instance with low and high demand patterns, leading to 10 tuning instances. Table 4 presents the parameter values used as default and Table 5 shows their values in tuning tests.

The cost obtained and execution time for each instance on each of the tests are presented in Table 6. The results show the trade-off between the quality of the solution and the change in execution time. Parameters of test 4 produce slightly better results than the default one but at the expense of the computing time, almost four times higher. Considering this trade-off, we keep the parameters as their default values in our computational experiments.

Table 4: Default values for the parameters

Parameters	Description	Value
$restart_{max}$	Maximum number of global iterations	$\|J\| \times 100$
p_{max}	Maximum number of perturbations	2
$Iter_{max}$	Number of iterations without improvement	15
$tabu_{min}$	Minimum length of the tabu list	10
$tabu_{max}$	Maximum length of the tabu list	30

Table 5: Values for parameter tuning tests

Parameters	Test (#): Value
$restart_{max}$	Test 1: $\|J\| \times 10$ Test 2: $\|J\|$
p_{max}	Test 3: 4 Test 4: 6
$Iter_{max}$	Test 5: 10 Test 6: 20
$tabu_{min}$	Test 7: 15 Test 8: 20
$tabu_{max}$	Test 9: 20 Test 10: 40

Table 6: Results of tuning tests

Instance\Test	Default	1	2	3	4	5	6	7	8	9	10
Instance	**Cost**										
TL12_LTL20_H	755.54	766.62	773.03	755.54	754.51	761.51	755.54	757.22	754.94	755.54	756.15
TL12_LTL20_L	690.29	700.23	763.23	694.21	688.17	702.48	692.21	700.03	701.95	698.85	700.23
TL16_LTL20_H	5095.30	5121.77	5136.30	5093.70	5098.26	5082.18	5095.68	5098.29	5101.00	5096.67	5100.05
TL16_LTL20_L	1887.77	1889.89	1980.16	1884.24	1886.34	1890.12	1887.91	1883.32	1889.89	1886.36	1892.93
TL20_LTL20_H	1358.62	1398.45	1433.06	1376.38	1374.00	1378.10	1382.94	1376.51	1362.31	1389.00	1393.31
TL20_LTL20_L	6078.08	6085.01	6205.93	6080.36	6074.07	6093.27	6081.47	6082.92	6089.26	6075.87	6073.67
TL4_LTL20_H	579.08	579.45	582.20	576.30	576.30	579.08	576.30	579.45	579.08	576.30	577.95
TL4_LTL20_L	477.99	487.76	511.08	474.95	462.61	471.65	477.99	475.08	473.72	462.61	477.99
TL8_LTL20_H	716.73	740.69	757.79	710.01	699.48	714.59	699.48	706.05	710.61	719.91	726.69
TL8_LTL20_L	571.89	584.68	620.82	571.04	571.39	571.39	571.04	571.39	571.39	576.95	571.39
Average	**1821.13**	**1835.45**	**1876.36**	**1821.67**	**1818.51**	**1824.44**	**1822.06**	**1823.03**	**1823.41**	**1823.80**	**1827.04**
Instance	**Time**										
TL12_LTL20_H	24	3	0	5	4	14	40	18	21	5	11
TL12_LTL20_L	9	1	0	52	28	13	26	1	27	28	26
TL16_LTL20_H	2	3	0	1	84	3	25	27	34	29	6
TL16_LTL20_L	9	4	0	21	82	18	43	41	31	9	41
TL20_LTL20_H	17	1	0	57	109	9	29	56	35	11	1
TL20_LTL20_L	25	6	0	7	122	27	21	19	51	24	28
TL4_LTL20_H	6	0	0	4	6	16	13	12	14	11	14
TL4_LTL20_L	8	2	0	18	22	3	2	22	8	6	16
TL8_LTL20_H	4	3	0	11	50	14	21	9	3	1	25
TL8_LTL20_L	24	2	0	38	10	9	5	7	27	30	17
Average	**13**	**3**	**0**	**21**	**52**	**13**	**22**	**21**	**25**	**15**	**18**

7.4. Performance of MS-ILS

For each combination of TL and LTL, we created four instances, see Table 2. A total of 160 instances are run five times, and the average results are presented in Table 7. The best solutions among the five runs are saved and the average of the best solutions of four instances for each combination of TL and LTL is shown under columns *Best Sol*.

To evaluate the performance of the proposed algorithm, we follow the definition provided by Laporte and Toth (2022) to compare the solutions obtained with the MS-ILS ($BestSol$) with the lower bound (LB) obtained by the commercial solver. Column $GAP_{UB}(\%)$ compares the average best solutions obtained by the MS-ILS with the LB obtained from the commercial solver, where $GAP_{UB}(\%) = 100 \times \dfrac{BestSol - LB}{LB}$. The next columns of cost, best time, and total time show the average cost, best time, and total time obtained over all runs and all instances for each combination of TL and LTL, respectively.

Table 7: Results of the proposed algorithm (MS-ILS)

		High demand pattern					Low demand pattern				
TL	LTL	Best Sol.	$GAP_{UB}(\%)$	Average cost	best time	total time	Best Sol.	$GAP_{UB}(\%)$	Average cost	best time	total time
4	4	319.72	0.00	319.72	0.60	6.26	279.40	0.00	280.03	0.63	6.25
4	8	322.61	0.00	322.65	2.70	10.80	322.25	0.00	322.25	2.31	10.49
4	12	455.93	3.67	456.03	2.85	16.90	379.98	7.26	380.15	2.89	16.99
4	16	593.12	10.88	594.57	9.41	24.87	448.06	17.55	450.40	5.48	25.12
4	20	542.90	17.65	545.69	15.69	32.34	459.37	13.39	460.13	11.31	33.83
8	4	423.03	0.00	423.03	0.72	9.16	404.15	0.00	404.15	1.11	9.56
8	8	495.00	0.00	495.00	1.52	14.93	506.45	0.00	506.85	2.94	15.43
8	12	574.21	5.08	574.53	6.57	22.20	523.26	2.24	524.68	8.89	23.41
8	16	600.08	7.09	603.35	11.65	30.72	604.17	6.57	606.71	10.50	30.82
8	20	779.22	16.77	785.42	22.54	38.16	618.05	12.54	619.14	15.20	41.34
12	4	566.40	0.05	566.40	0.90	13.40	566.22	0.00	566.22	1.18	12.98
12	8	642.32	0.00	643.01	4.01	19.21	671.75	0.06	672.80	5.21	20.01
12	12	786.52	3.44	787.67	10.69	27.39	784.90	2.46	791.04	11.28	27.80
12	16	899.48	6.87	904.12	12.41	36.18	703.82	6.54	707.47	15.49	35.48
12	20	957.59	13.75	960.39	19.85	44.36	821.47	9.86	825.79	25.29	47.10
16	4	726.64	0.00	727.36	5.68	18.24	769.67	0.07	771.06	7.40	17.33
16	8	848.56	0.33	850.47	7.63	23.70	745.16	0.06	746.74	6.41	24.49
16	12	809.32	2.31	812.40	13.36	34.34	834.88	0.00	835.58	7.58	29.28
16	16	1061.51	8.26	1068.91	25.73	46.27	898.49	7.58	900.82	21.69	49.37
16	20	1055.25	12.18	1062.08	38.83	62.68	979.63	9.32	984.31	33.06	61.50
Global average		**672.97**	**5.42**	**675.14**	**10.67**	**26.61**	**616.05**	**4.77**	**617.81**	**9.79**	**26.93**

As it can be observed from Table 7 and compared to the results shown in Table 3, in general, the proposed algorithm finds solutions as good as the ones obtained by the commercial solver, with an average of 0.20% and 0.23% deviation from the upper bounds for instances with high and low demand pattern, respectively. However, these solutions are obtained in less than 27 seconds compared to 12,425 and 11,432 seconds spent by the commercial solver for instances with high and low demand patterns. If we consider the time the best solutions are obtained by the proposed algorithms, the comparisons becomes even more interesting, as on average the best solutions is obtained in no longer than 11 seconds.

7.5. Testing ILS on single truck and trailer routing problem with satellite depots instances

As the STTRPSD is close to our problem, and to evaluate the performance of the proposed metaheuristic, we adapted our algorithm to solve the instances of this problem from the literature. In this case, we must use only one road train and visit more than one intermediate terminal. We also assign only one regular truck to each intermediate terminal. The calculation of the objective function is also altered to consider the order in which the intermediate terminals appear in the solution.

Villegas et al. (2010) generated a set of 32 instances characterized by the number of customers ranging from 25 to 200 and the number of intermediate depots ranging from from 5 to 20. Customer demands follow a uniform distribution in the interval $[1, 200]$. The distribution of customers and trailer points is clustered (c) or random (rd). The capacity of trucks is either 1,000 or 2,000. Euclidean coordinates are randomly selected in a square grid of 100×100. More information on these instances can be found in Villegas et al. (2010).

Table 8 compares the results obtained by our proposed metaheuristic with those from Villegas et al. (2010) and Accorsi and Vigo (2020). The first four columns provide information on the STTRPSD instances. We solved each instance five times and, as in Villegas et al. (2010) and Accorsi and Vigo (2020), we report the best solution obtained (Best), average solution (Avg), and the average execution time (Time, in seconds). Then, methods from Villegas et al. (2010) including enhanced cluster-first, route second (labelled CFRS), CFRS and VND (labelled CFRS+VND), and iterated route-first, cluster-second (labeled IRFCS) are compared. For IRFCS, the best and average solutions obtained from ten runs are shown in columns Best and Avg. Finally, the last two columns present the best and average solutions obtained by the hybrid metaheuristic proposed by Accorsi and Vigo (2020).

As Table 8 shows, from 16 instances with less than 100 customers, we improved the best known solution from the literature for two instances, identified in boldface. For larger instances with more than 100 customers, the performance of our method slightly improves. On these 16 larger instances, we obtained thirteen times the same solution as the ones reported in Accorsi and Vigo (2020) and improved the best known solution from the literature in three cases, identified in boldface. Our adapted algorithm outperforms all those presented in Villegas et al. (2010) as confirmed by the global averages shown at the bottom of the table. To have the same basis of comparison for all methods, we also report the average of percentage Gap_{UB}, where the best known solution (BKS) for each instance is used as the reference solutions. Therefore, $GAP_{UB}(\%) = 100 \times \dfrac{UB - BKS}{BKS}$, where UB is the solution obtained by the algorithm being compared.

Table 8: Performance comparison on STTRPSD instances

				MS-ILS			**Three heuristics** ([Villegas et al., 2010](#))				**Hybrid metaheuristic** (**Accorsi and Vigo, 2020**)	
Instance	n	p	Type	Best	Avg.	Time	CFRS	CFRS+VND	IRFCS (Best)	IRFCS (Avg)	Best	Avg
1	25	5	c	405.46	405.46	93.73	444.08	405.46	420.34	427.48	405.46	405.46
2	25	5	c	374.79	388.26	29.95	444.08	391.62	390.6	398.56	374.79	374.79
3	25	5	rd	584.03	584.03	73.70	696.73	585.96	596.17	618.61	584.03	584.03
4	25	5	rd	508.48	521.17	15.47	640.01	526.27	530.48	548.32	508.48	508.48
5	25	10	c	386.45	396.13	69.51	460.28	386.45	398.41	404.86	386.45	386.45
6	25	10	c	380.86	381.32	20.86	460.28	386.45	391.43	401.46	380.86	380.86
7	25	10	rd	573.96	578.10	64.39	789.7	582.64	597.13	613.02	573.96	573.96
8	25	10	rd	506.37	508.85	57.81	789.7	582.64	521.67	542.26	506.37	506.37
9	50	5	c	**565.54**	572.04	144.48	625.67	583.41	641.15	646.42	583.07	583.07
10	50	5	c	516.98	545.05	460.44	574.17	560.17	594.72	608.57	516.98	516.98
11	50	5	rd	**831.11**	831.11	510.53	1177.25	870.51	994.62	1012.4	870.51	870.51
12	50	5	rd	766.03	766.03	102.49	980.57	787.79	895.6	907.91	766.03	766.03
13	50	10	c	387.83	397.79	199.29	471.43	387.83	424.53	433.5	387.83	387.83
14	50	10	c	367.01	385.25	185.32	460.36	381.32	415.45	421.57	367.01	367.01
15	50	10	rd	811.28	865.52	327.81	1034.77	847.49	892.42	931.37	811.28	811.28
16	50	10	rd	731.53	753.62	142.48	1013.2	758.95	855.75	874.4	731.53	731.53
17	100	10	c	**607.56**	615.54	643.06	705.19	640.01	724.13	742.97	614.02	614.02
18	100	10	c	547.44	560.17	425.20	665.76	555.31	679.7	691.22	547.44	547.44
19	100	10	rd	**1169.22**	1181.50	499.68	1544.01	1416.6	1569.15	1593.06	1271.81	1271.81
20	100	10	rd	1097.28	1128.10	418.90	1290.79	1167.97	1378.2	1408.73	1097.28	1097.28
21	100	20	c	642.61	713.59	173.81	820	668.04	768.01	781.21	642.61	642.61
22	100	20	c	581.56	624.59	203.63	808.61	643.16	692.03	714.43	581.56	581.56
23	100	20	rd	1143.10	1211.59	160.02	1392.01	1192.83	1374.35	1410.23	1143.1	1143.1
24	100	20	rd	**1060.75**	1136.45	218.00	1342.1	1138.84	1321.92	1348.44	1060.86	1063.3
25	200	10	c	819.97	938.60	4680.36	1004.8	849.63	1032.04	1049.39	819.97	819.97
26	200	10	c	710.69	765.17	3386.42	878.59	734.63	936.67	949.52	710.69	711.19
27	200	10	rd	1755.46	1841.66	5386.00	2391.08	2026.04	2305.69	2322.63	1755.46	1755.46
28	200	10	rd	1445.94	1584.38	2892.65	1951.77	1515.01	2028.96	2050.88	1445.94	1445.94
29	200	20	c	907.19	1106.18	901.62	1098.7	950.21	1134.92	1152.02	907.19	907.19
30	200	20	c	814.45	1040.68	1001.53	1036.27	862.35	1040.35	1056.52	814.45	814.45
31	200	20	rd	1610.63	1750.11	1042.91	2251.76	1691.43	2142.39	2167.92	1610.63	1610.63
32	200	20	rd	1413.32	1539.91	758.20	2019.44	1559.43	1956.08	1975.18	1413.32	1413.32
Average				**782.03**	**828.04**	**740.12**	**1008.22**	**832.39**	**957.66**	**975.16**	**787.22**	**787.31**
Average Gap_{UB} (%)				**0.00**	**5.15**		**26.93**	**5.36**	**18.19**	**20.56**	**0.55**	**0.56**

After having the results compared to the best known solutions from the literature and since multi-start frameworks have already been proposed in Villegas et al. (2010) to solve the benchmark instances, in what follows, we compare the results with other multi-start algorithms. In Table 9, we refer to our MS-ILS algorithm as the Proposed Algo. where we show the best, worst, and average solutions obtained for each instance. The results are compared against the ones from two multi-start methods, reported in Villegas et al. (2010). These methods are multi start evolutionary local search (ELS) and their MS-ILS. Our algorithm clearly produces better solutions than these two methods. However, the algorithms proposed by Villegas et al. (2010) are more stable with respect to average and worst solutions. The same as for the previous table, the average percentage Gap_{UB}, where the BKS for each instance is used as the reference solutions, is reported.

Table 9: Performance comparison with other multi-start algorithms on STTRPSD instances

	Proposed Algo.			**MS-ELS**			**MS-ILS**		
Instance	**Best**	**Worst**	**Avg.**	**Best**	**Worst**	**Avg.**	**Best**	**Worst**	**Avg.**
1	405.46	405.46	405.46	405.46	405.46	405.46	405.46	405.46	405.46
2	374.79	391.62	388.26	374.79	374.79	374.79	374.79	374.79	374.79
3	584.03	584.03	584.03	584.03	584.03	584.03	584.03	584.03	584.03
4	508.48	524.34	521.17	508.48	508.48	508.48	508.48	508.48	508.48
5	386.45	401.07	393.80	386.45	386.45	386.45	386.45	386.45	386.45
6	380.86	381.43	381.32	380.86	380.86	380.86	380.86	380.86	380.86
7	573.96	579.19	576.91	573.96	573.96	573.96	573.96	573.96	573.96
8	506.37	509.55	508.85	506.37	506.37	506.37	506.37	506.37	506.37
9	**565.54**	572.13	569.54	583.07	583.41	583.1	583.07	583.07	583.07
10	516.98	552.77	544.68	516.98	516.98	516.98	516.98	516.98	516.98
11	**831.11**	831.11	831.11	870.51	870.51	870.51	870.51	870.51	870.51
12	**766.03**	766.03	766.03	766.03	766.03	766.03	766.03	766.03	766.03
13	387.83	403.55	395.29	387.83	387.83	387.83	387.83	387.83	387.83
14	367.01	392.69	384.70	367.01	367.01	367.01	367.01	367.01	367.01
15	811.28	892.06	857.37	811.28	811.28	811.28	811.28	811.28	811.28
16	731.53	764.15	755.34	731.53	731.53	731.53	731.53	731.53	731.53
17	**607.56**	621.94	615.71	614.2	614.31	614.3	614.02	615.32	614.41
18	547.44	564.22	559.92	547.44	548.11	547.64	547.44	548.11	547.57
19	**1169.22**	1189.28	1181.50	1275.76	1285.38	1280.65	1280.02	1286.14	1282.79
20	1097.28	1143.82	1127.10	1097.28	1097.28	1097.28	1097.28	1103.43	1097.9
21	642.61	746.06	704.52	642.61	643.93	642.79	642.61	642.61	642.61
22	581.56	636.55	617.03	581.56	583.71	581.78	581.56	583.71	582.18
23	1143.10	1245.05	1206.48	1143.1	1151.29	1147.11	1143.1	1150.34	1146.66
24	1060.75	1161.90	1132.73	1060.75	1066.54	1064.04	1060.75	1064.99	1062.84
25	**819.97**	986.54	935.02	827.1	836.39	830.99	822.52	835.02	828.37
26	**710.69**	783.86	762.26	715.37	729.98	723.17	714.33	725.99	719.99
27	**1755.46**	1881.82	1827.33	1761.1	1807.95	1787.63	1763.3	1805.7	1783.24
28	1445.94	1633.42	1569.29	1454.9	1475.95	1461.06	1445.94	1470.7	1458.15
29	**907.19**	1173.85	1105.17	912.87	920.76	916.17	909.46	923.22	913.51
30	**814.45**	1101.65	1040.68	815.51	824.51	820.64	820.67	824.84	822.19
31	**1610.63**	1812.00	1727.43	1620.47	1648.79	1632.1	1614.18	1649.12	1631.61
32	1413.32	1577.25	1521.14	1420.45	1449.77	1432.01	1413.32	1438.97	1424.77
Average	**782.03**	**850.33**	**828.04**	**788.91**	**794.99**	**791.69**	**788.29**	**794.46**	**791.04**
Average Gap_{UP} **(%)**	**0.00**	**7.55**	**5.15**	**0.7**	**1.20**	**0.93**	**0.65**	**1.16**	**0.87**

7.6. Sensitivity analysis on the number of trailers loaded on road trains

In this section, we provide managerial insights on the use of road trains. In our case study, based on the Canadian context, we assumed that each road train could only load two trailers. However, in other countries, such as Australia, the limit is three trailers. Therefore, we conduct the following analyses to understand how much improvement can be obtained if the two-trailer regulation is extended.

In the mathematical model presented in Section 4, constraints (3) set the number of trailers to be loaded on each road train to two. First, we limit the number to three by modifying the right-hand side of these constraints as presented below (constraints (14)).

$$\sum_{k \in K} y_{tk} = 3w_t \quad \forall t \in T \backslash \{1\} \tag{14}$$

Then, we study how the model optimizes the number of trailers loaded on a road train when having more flexibility. In this case, we modify the model by replacing constraints (3) with constraints (15), as shown below.

$$\sum_{k \in K} y_{tk} \leq 3w_t \quad \forall t \in T \backslash \{1\} \tag{15}$$

The results are presented in Tables 10 and 11 for instances with high and low demand patterns, respectively. The first two columns in these tables provide the instance sets information. Then the average of upper bounds (UB), the optimality gap ($Gap\%$), and the execution time in seconds ($Sec.$) are reported for using two trailers, three trailers, and either two or three trailers. We expect that the model with the most flexibility obtains the best solutions. Moreover, we expect that increasing the number of trailers will lead to cost minimization since the fixed cost remains the same.

From the results shown in Tables 10 and 11, we observe that, in general, changing the limit on the number of loaded trailers from two to three, makes the problem more difficult to solve, proved by the increase in the average gap and the execution time. However, this increase in the number of loaded trailers will result in 6.82% and 6.24% global average cost reduction for high and low demand instances, respectively.

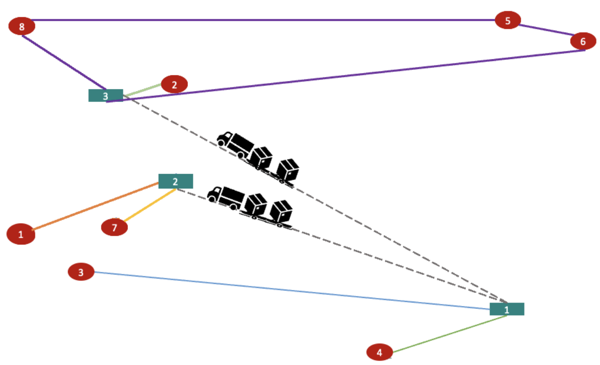
Table 10: Comparison of the results for different number of trailers for high demand instances

		Two trailers			Three trailers			Two or three trailers		
TL	LTL	UB	Gap%	Sec.	UB	Gap%	Sec.	UB	Gap%	Sec.
4	4	319.72	0.00	3	297.05	0.00	2	294.97	0.00	3
4	8	322.60	0.00	1136	308.82	0.00	1902	307.28	0.00	3826
4	12	455.93	3.67	17814	431.68	9.09	21600	431.68	9.21	21600
4	16	592.92	10.84	21600	545.73	13.11	21600	543.26	12.55	21600
4	20	539.80	16.96	21601	515.79	23.86	21601	515.89	22.75	21601
8	4	423.03	0.00	2	394.64	0.00	3	394.64	0.00	4
8	8	495.00	0.00	334	472.41	0.00	1882	471.27	0.00	1226
8	12	574.21	5.08	21600	557.00	8.87	21600	551.93	9.21	21600
8	16	598.22	6.76	21600	566.79	7.78	21600	566.79	7.84	21600
8	20	769.31	15.35	21600	721.40	19.12	21600	720.60	21.39	21604
12	4	566.09	0.00	8	523.10	0.00	9	523.10	0.00	38
12	8	642.32	0.00	756	595.75	0.00	805	595.75	0.00	770
12	12	786.52	3.44	21522	727.50	6.21	21600	727.50	6.18	21600
12	16	888.30	5.60	21600	829.38	9.49	21600	829.38	9.30	21600
12	20	952.90	13.24	21600	878.55	17.44	21602	878.51	19.82	21606
16	4	726.64	0.00	13	676.63	0.00	33	675.17	0.00	27
16	8	845.72	0.00	485	783.32	0.00	1952	783.26	0.00	1281
16	12	807.45	2.10	17428	750.65	2.02	16295	748.34	2.37	20192
16	16	1080.79	10.25	21600	973.83	6.43	21601	973.83	8.31	21600
16	20	1044.32	11.09	21601	965.14	14.82	21601	962.96	14.21	21602
Global average		**671.59**	**5.22**	**12695**	**625.76**	**6.91**	**13024**	**624.81**	**7.16**	**13249**

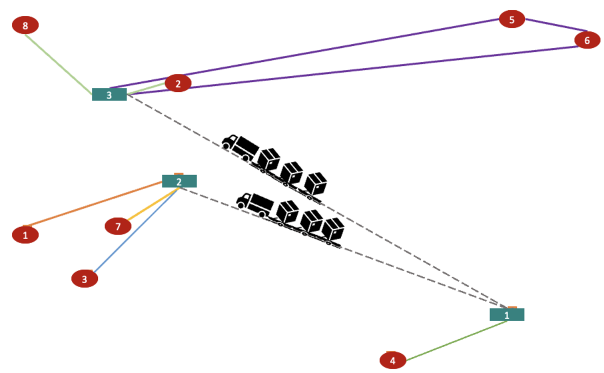
Table 11: Comparison of the results for different number of trailers for low demand instances

		Two trailers			Three trailers			Two or three trailers		
TL	LTL	Sol.	Gap%	Sec.	Sol.	Gap%	Sec.	Sol.	Gap%	Sec.
4	4	279.40	0.00	1	274.77	0.00	3	272.11	0.00	2
4	8	322.25	0.00	220	313.14	0.00	1176	310.98	0.00	750
4	12	379.50	7.14	21600	351.92	10.09	21600	351.33	9.57	19619
4	16	448.06	17.55	21600	422.80	17.07	21600	422.80	18.21	21600
4	20	459.75	13.49	21600	442.19	15.06	21600	437.35	16.11	21600
8	4	404.15	0.00	2	379.02	0.00	4	376.40	0.00	4
8	8	506.45	0.00	491	479.61	0.00	2669	479.61	0.00	1125
8	12	523.26	2.24	18644	491.48	4.37	21600	491.48	5.13	21600
8	16	604.16	6.57	21600	568.86	7.96	21600	568.86	8.03	21600
8	20	618.00	12.52	21600	584.32	11.48	21600	582.28	11.33	21600
12	4	566.21	0.00	6	520.74	0.00	10	520.74	0.00	10
12	8	671.33	0.00	238	619.86	0.79	7420	619.86	0.00	3654
12	12	781.36	1.99	13145	733.85	4.54	16652	733.85	4.44	16512
12	16	697.40	5.59	21600	646.93	7.14	21600	646.93	9.17	21600
12	20	808.46	8.09	21600	764.20	8.61	21600	764.20	8.84	21601
16	4	769.13	0.00	14	715.97	0.00	23	715.97	0.00	27
16	8	744.66	0.00	727	683.38	0.00	3765	683.38	0.00	3988
16	12	834.88	0.00	743	788.61	0.00	5451	786.81	0.00	2754
16	16	894.56	7.09	21600	831.57	7.52	21601	831.09	7.44	21601
16	20	973.88	8.63	21600	906.26	9.92	21601	906.02	9.30	21601
Global average		**614.34**	**4.54**	**11432**	**575.97**	**5.23**	**12659**	**575.10**	**5.38**	**12142**

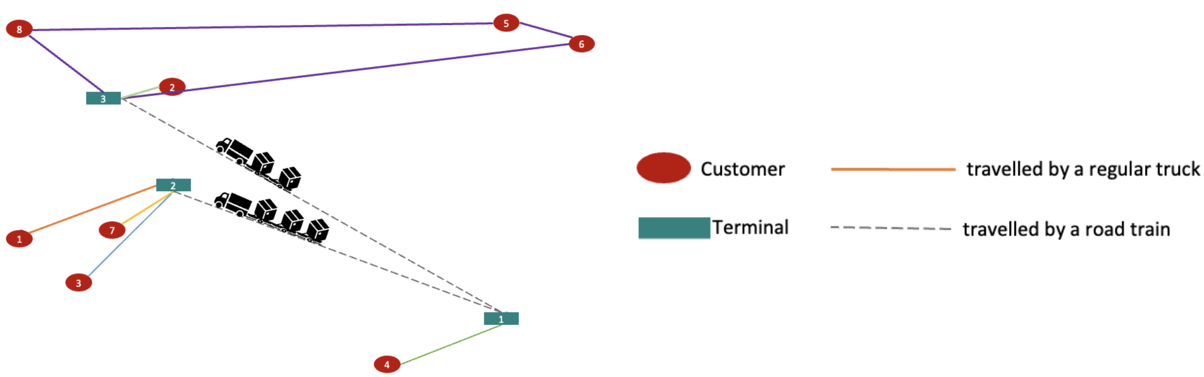
As expected, when the model is flexible in choosing the number of loaded trailers, i.e., either two or three, the cost is further reduced, but the optimality gap and execution time are increased. To better observe these effects and how the solution changes, in Figure 2, we show the optimal solutions obtained for an instance with four TLs and four LTLs with a high-demand pattern. In all three solutions, a road train is sent from the main terminal (terminal 1) to the other two (terminals 2 and 3). Also, customer 4 is always visited from the main terminal, customers 1 and 7 from terminal 2, and customers 2, 5, 6, 8 from terminal 3. However, when only two trailers are loaded on a road train, customer 3 will be visited from terminal 1. The other difference is in how the routes are built, in the flexible case, as is the case with two loaded trailers, the model chooses to send two trailers to terminal 3, and creates a route to server customers 5, 6, and 8 whereas, in the case with three loaded trailers, three routes are generated to serve customers, 2, 5, 6, and 8.



(a) with two trailers
Solution: 276.55



(b) with three trailers
Solution: 256.62



(c) with two or three trailers
Solution: 248.30

Figure 2: Optimal solutions for instance TL4-LTL4-H-A with different number of trailers per road train.

8. Conclusion

This paper introduces and solves the road train optimization problem with load assignment (RTOP–LA). RTOP–LA deals with selecting intermediate terminals for road trains and making routing decisions for trucks from all terminals to customers. Trailers are initially located in an origin terminal, loaded with customer demands. These trailers can be carried by regular trucks sent directly to customers from the origin terminal or the other option is to connect two trailers to a road train and send them to an intermediate terminal where they will be dismantled. These trailers are then assigned to regular trucks and sent to final customers. We also considered two types of orders: full truckload (TL) and less-than-truckload (LTL). A new model for RTOP–LA has been introduced and a set of 160 instances based on our collaboration with a trucking company is developed. We also developed an algorithm based on the multi-start iterated local search (MS-ILS) metaheuristic to obtain quality solutions for larger instances.

In order to have an independent basis of comparison, we have evaluated the MS-ILS over the benchmark instances of the single truck and trailer routing problem with satellite depots (STTRPSD) against the best algorithms available. Out of 32 instances, we report five new best known solutions. Finally, our analyses show that additional gains can be obtained by having more flexibility in the choice of the number of trailers loaded on a road train.

In future work, we suggest applying this problem to other minimization objectives, such as delivery time and the number of movements in loading and unloading demands. Moreover, this problem can be extended considering uncertainties, for example, of demands and delivery or service time. Scheduling deliveries considering the time required to load/unload trailers and to dismantle road trains at the terminals also opens doors to other interesting and practical problems inspired by our work in this paper.

Acknowledgments

This project was partly funded by the Canadian Natural Sciences and Engineering Research Council (NSERC) under grants 2020-00401 and 2018-03712. We also thank our industrial partner for providing us with the data and all technical specifications. Finally, we would like to thank the associate editor and the two anonymous referees for their valuable comments and suggestions on an earlier version of this paper.

References

Absi, N., Cattaruzza, D., Feillet, D., Housseman, S., 2017. A relax-and-repair heuristic for the swap-body vehicle routing problem. Annals of Operations Research 253, 957–978.

Accorsi, L., Vigo, D., 2020. A hybrid metaheuristic for single truck and trailer routing problems. Transportation Science 54, 1351–1371.

Accorsi, L., Vigo, D., 2021. A fast and scalable heuristic for the solution of large-scale capacitated vehicle routing problems. Transportation Science 55, 832–856.

Atefi, R., Salari, M., Coelho, L.C., Renaud, J., 2018. The open vehicle routing problem with decoupling points. European Journal of Operational Research , 316–327.

Baldacci, R., Mingozzi, A., Roberti, R., Calvo, R.W., 2013. An exact algorithm for the two-echelon capacitated vehicle routing problem. Operations Research 61, 298–314.

Bhoopalam, A.K., Agatz, N., Zuidwijk, R., 2018. Planning of truck platoons: A literature review and directions for future research. Transportation research part B: Methodological 107, 212–228.

Boysen, N., Fedtke, S., Schwerdfeger, S., 2021. Last-mile delivery concepts: a survey from an operational research perspective. OR Spectrum 43, 1–58.

Boysen, N., Scholl, J., Stephan, K., 2017. When road trains supply freight trains: scheduling the container loading process by gantry crane between multi-trailer trucks and freight trains. OR spectrum 39, 137–164.

Cabrera, N., Cordeau, J.F., Mendoza, J.E., 2022. The doubly open park-and-loop routing problem. Computers & Operations Research 143, 105761.

Canada, T., 2020. Transportation in canada 2019. URL: https://tc.canada.ca/fr/services-generaux/transparence/gestion-rapports-ministeriels/rapports-annuels-transports-canada/transports-canada-2019. available online (accessed on June 30, 2021).

Chao, I.M., 2002. A tabu search method for the truck and trailer routing problem. Computers & Operations Research 29, 33–51.

Chen, Y., Peterson, A.W., Zhang, C., Ahmadian, M., 2019. A simulation-based comparative study on lateral characteristics of trucks with double and triple trailers. International journal of vehicle safety 11, 136–157.

Crainic, T.G., Kim, K.H., 2007. Intermodal transportation. Handbooks in operations research and management science 14, 467–537.

Cuda, R., Guastaroba, G., Speranza, M.G., 2015. A survey on two-echelon routing problems. Computers & Operations Research 55, 185–199.

Darvish, M., Archetti, C., Coelho, L.C., Speranza, M.G., 2019. Flexible two-echelon location routing problem. European Journal of Operational Research 277, 1124–1136.

Derigs, U., Pullmann, M., Vogel, U., 2013. Truck and trailer routing—problems, heuristics and computational experience. Computers & Operations Research 40, 536–546.

Dessouky, Y., Tsao, H.S.J., Patel, M.H., Zeta, J.B., Zhou, L., 2007. A simulation study of the productivity of large trucks with shorter trailers. International Journal of Industrial and Systems Engineering 2, 261–285.

Dondo, R., Méndez, C.A., Cerdá, J., 2011. The multi-echelon vehicle routing problem with cross docking in supply chain management. Computers & Chemical Engineering 35, 3002–3024.

Drexl, M., 2013. Applications of the vehicle routing problem with trailers and transshipments. European Journal of Operational Research 227, 275–283.

Gouvernement du Québec, 2019. Inventaire québécois des émissions de gaz à effet de serre en 2017 et leur évolution depuis 1990. Technical Report ISBN 978-2-550-85631-3. Gouvernement du Québec.

Guastaroba, G., Speranza, M.G., Vigo, D., 2016. Intermediate facilities in freight transportation planning: a survey. Transportation Science 50, 763–789.

Hansen, P., Mladenović, N., 2014. Variable neighborhood search, in: Burke E., K.G. (Ed.), Search methodologies. Springer, pp. 313–337.

Huber, S., Geiger, M.J., 2017. Order matters–a variable neighborhood search for the swap-body vehicle routing problem. European Journal of Operational Research 263, 419–445.

Jagelčák, J., Kiktová, M., Frančák, M., Marienka, P., 2019. The possibilities of using longer and heavier vehicle combinations in slovakia. Transportation research procedia 40, 271–278.

Keaton, M.H., 1991. Service-cost tradeoffs for carload freight traffic in the us rail industry. Transportation Research Part A: General 25, 363–374.

Lai, X., Lü, Z., 2013. Multistart iterated tabu search for bandwidth coloring problem. Computers & Operations Research 40, 1401–1409.

Laporte, G., 1992. The traveling salesman problem: An overview of exact and approximate algorithms. European Journal of Operational Research 59, 231–247.

Laporte, G., Nobert, Y., 1987. Exact algorithms for the vehicle routing problem, in: North-Holland Mathematics Studies. Elsevier. volume 132, pp. 147–184.

Laporte, G., Toth, P., 2022. A gap in scientific reporting. 4OR 20, 169–171.

Laumanns, N., Laumanns, M., Neunzig, D., 2001. Multi-objective design space exploration of road trains with evolutionary algorithms, in: International Conference on Evolutionary Multi-Criterion Optimization, Springer. pp. 612–623.

Lightstone, A., Belony, B., Cappuccilli, J.F., 2021. Understanding goods movement in canada: Trends and best practices.

Lin, S.W., Vincent, F.Y., Chou, S.Y., 2009. Solving the truck and trailer routing problem based on a simulated annealing heuristic. Computers & Operations Research 36, 1683–1692.

Lourenço, H.R., Martin, O.C., Stützle, T., 2019. Iterated local search: Framework and applications, in: Gendreau M., P.J. (Ed.), Handbook of Metaheuristics. Springer, Cham, pp. 129–168.

Ministère des Transports, 2020. Guide regarding the Special Road Train Operating Permits Regulation. Technical Report ISBN 978-2-550-86772-2. Gouvernement du Québec.

Miranda-Bront, J.J., Curcio, B., Méndez-Díaz, I., Montero, A., Pousa, F., Zabala, P., 2017. A cluster-first route-second approach for the swap body vehicle routing problem. Annals of Operations Research 253, 935–956.

Nguyen, V.P., Prins, C., Prodhon, C., 2012. A multi-start iterated local search with tabu list and path relinking for the two-echelon location-routing problem. Engineering Applications of Artificial Intelligence 25, 56–71.

Penna, P.H.V., Subramanian, A., Ochi, L.S., 2013. An iterated local search heuristic for the heterogeneous fleet vehicle routing problem. Journal of Heuristics 19, 201–232.

Perboli, G., Tadei, R., Vigo, D., 2011. The two-echelon capacitated vehicle routing problem: Models and math-based heuristics. Transportation Science 45, 364–380.

Semet, F., 1995. A two-phase algorithm for the partial accessibility constrained vehicle routing problem. Annals of Operations Research 61, 45–65.

Sluijk, N., Florio, A.M., Kinable, J., Dellaert, N., Van Woensel, T., 2023. Two-echelon vehicle routing problems: A literature review. European Journal of Operational Research 304, 865–886.

Smith, A., Toth, P., Bam, L., van Vuuren, J.H., 2021. A multi-tiered vehicle routing problem with global cross-docking. Computers & Operations Research , 105526.

Subramanian, A., Drummond, L.M., Bentes, C., Ochi, L.S., Farias, R., 2010. A parallel heuristic for the vehicle routing problem with simultaneous pickup and delivery. Computers & Operations Research 37, 1899–1911.

Subramanyam, A., Repoussis, P.P., Gounaris, C.E., 2020. Robust optimization of a broad class of heterogeneous vehicle routing problems under demand uncertainty. INFORMS Journal on Computing 32, 661–681.

Todosijević, R., Hanafi, S., Urošević, D., Jarboui, B., Gendron, B., 2017. A general variable neighborhood search for the swap-body vehicle routing problem. Computers & Operations Research 78, 468–479.

Villegas, J.G., Prins, C., Prodhon, C., Medaglia, A.L., Velasco, N., 2010. Grasp/vnd and multi-start evolutionary local search for the single truck and trailer routing problem with satellite depots. Engineering Applications of Artificial Intelligence 23, 780–794.