Computers and Operations Research 118 (2020) 104919

Contents lists available at ScienceDirect

Computers and Operations Research

journal homepage: www.elsevier.com/locate/cor

The two-echelon vehicle routing problem with covering options: City logistics with cargo bikes and parcel lockers

David L.J.U. Enthoven, Bolor Jargalsaikhan, Kees Jan Roodbergen, Michiel A.J. uit het Broek*, Albert H. Schrotenboer*

Department of Operations, Faculty of Economics and Business, University of Groningen, the Netherlands

A R T I C L E I N F O

Article history:
Received 27 March 2019
Revised 12 December 2019
Accepted 12 February 2020
Available online 13 February 2020

Keywords:
Two-echelon vehicle routing
Location routing
Cargo bikes
Parcel lockers
City logistics
Sustainable logistics

A B S T R A C T

We introduce the two-echelon vehicle routing problem with covering options (2E-VRP-CO). This problem arises in sustainable applications for e-commerce and city distribution. In the first echelon, trucks depart from a single depot and transport goods to two types of locations. At covering locations, such as parcel lockers, customers can pick up goods themselves. At satellite locations, goods are transferred to zero-emission vehicles (such as cargo bikes) that deliver to customers. If desired, customers can indicate their choice for delivery. The 2E-VRP-CO aims at finding cost-minimizing solutions by selecting locations and routes to serve all customers. We present a compact mixed integer programming formulation and an efficient and tailored adaptive large neighborhood search heuristic that provides high-quality, and often optimal, solutions to the 2E-VRP-CO. The 2E-VRP-CO has as special cases the two-echelon vehicle routing problem, and the simultaneous facility location and vehicle routing problem without duration constraints. On these special cases, for which our heuristic predominantly solves the established benchmark instances either to optimality or to the best-known solution, our heuristic finds three new best-known solutions. Moreover, we introduce a new set of benchmark instances for the 2E-VRP-CO and provide managerial insights when distribution via both satellite and covering locations is most beneficial. Our results indicate that customers in the same area are best-served either via cargo-bikes or parcel lockers (i.e., not both), and that the use of parcel lockers has a great potential to reduce driving distance.

1. Introduction

Municipalities are keen to reduce the amount of traffic congestion in their inner cities in order to improve the livability (Demir et al., 2015). As a result, several regulations have been implemented to limit the number of trucks in the inner city and to promote the usage of zero-emission vehicles such as cargo bikes or electric vehicles for last-mile delivery (Cattaruzza et al., 2017). Still, the number of trucks is increasing due to the growing popularity of e-commerce and the desire for faster delivery (Savelsbergh and Van Woensel, 2016). In order to reduce the number of trucks used for inner-city transportation, alternative approaches for the last-mile delivery have been proposed. One option is to use intermediate locations such as *satellite locations*. Here, parcels delivered by trucks are transferred to zero-emission vehicles such as cargo bikes that are small, manoeuvrable, and well-equipped to perform at-home deliveries in densely populated areas.

Another option is to introduce *covering locations* where trucks deliver parcels and nearby customers collect these parcels themselves, incurring exogenously given customer-specific connection costs (Arnold et al., 2018; Deutsch and Golany, 2018). Examples of such covering locations include self-accessible lockers at train or bus stations, local retail shops in small neighbourhoods, and other locations that are already visited frequently such that the resulting amount of additional traffic is kept to a minimum. The connection costs reflect the customers' preferred delivery method; customers who prefer to be served via a covering location have zero connection costs, while sufficiently high connection costs reflect the customers that prefer being served via cargo bikes.

We study how to effectively integrate the use of satellite and covering locations, which we jointly call intermediate locations. In other words, we consider the two-echelon vehicle routing problem with covering options (2E-VRP-CO), where goods are transshipped from a central depot to intermediate locations in the first echelon. In the second echelon, customers are serviced via covering loca-

* Corresponding author.
E-mail addresses: a.j.uit.het.broek@rug.nl (M.A.J. uit het Broek), a.h.schrotenboer@rug.nl (A.H. Schrotenboer).

https://doi.org/10.1016/j.cor.2020.104919

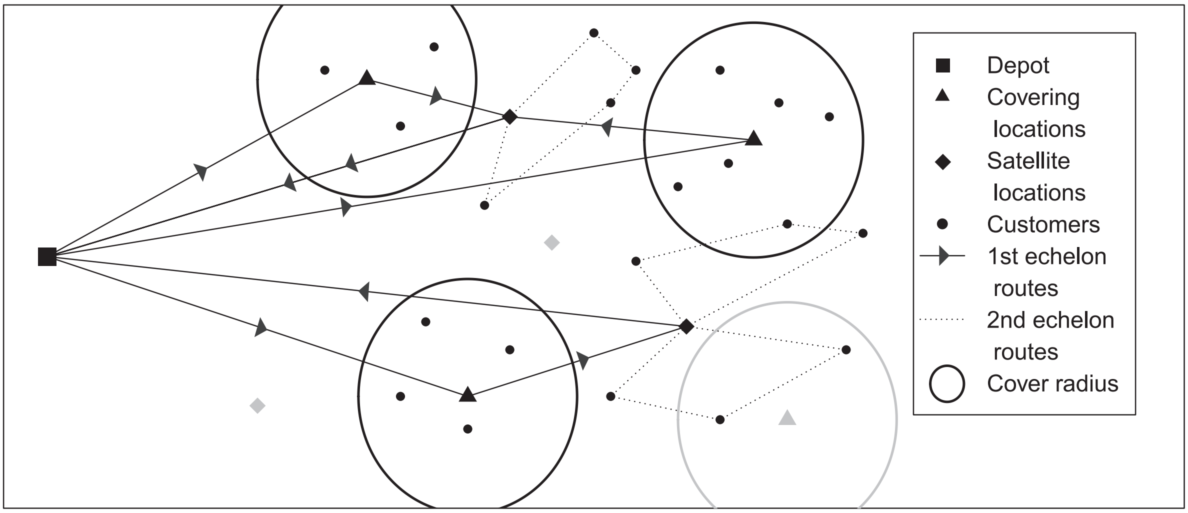


Fig. 1. Example of a solution for the 2E-VRP-CO.

tions or via a satellite location with cargo bikes. Fig. 1 provides a typical solution to the 2E-VRP-CO. It can be seen that each customer is either serviced by one of the second-echelon routes (dotted routes) or it is covered by one of the opened covering locations (black circles).

In the first echelon, we allow multiple trucks to deliver parcels to an intermediate location, i.e., the amount of parcels may be split between different trucks for intermediate locations. However, there is no split delivery for a customer in the second echelon. The goal of 2E-VRP-CO is to minimize the total day-to-day operational costs, which consist of routing costs for the trucks and cargo bikes and the incurred connections costs via using covering locations.

We develop a mixed integer programming (MIP) formulation for the 2E-VRP-CO, and show that it extends both the two-echelon vehicle routing problem (Breunig et al., 2016; Hemmelmayr et al., 2012) and the simultaneous facility location and vehicle routing problem without duration constraints (SFL-VRP, Veenstra et al., 2018). With the MIP formulation, we are able to solve relatively small instances to optimality. In order to provide high-quality solutions to practically-sized instances, we develop an adaptive large neighborhood search (ALNS) heuristic which has demonstrated its performance on related vehicle routing problems, see e.g., Grangier et al. (2016) and Breunig et al. (2019). We show that, on a set of newly developed benchmark instances, our ALNS provides high quality solutions. Moreover, our ALNS appears to be efficient in solving the aforementioned special cases as we provide two new best-known solutions for the 2E-VRP and one for the SFL-VRP.

The two-echelon vehicle routing problem is an extension of the classical vehicle routing problem where delivery from the central warehouse to the customer is done via satellite locations. We refer to the surveys by Cuda et al. (2015) and Guastaroba et al. (2016) for an overview of the 2E-VRP literature. There are many different variations of 2E-VRP coming from practice, for example Grangier et al. (2016) and Anderluh et al. (2017) consider the 2E-VRP with time-synchronization constraints and Wang et al. (2017) include environmental aspects. However, to the best of the authors' knowledge, this particular configuration of two-echelon routing with covering options has not been investigated before. Another related problem is two-echelon location routing problem (2E-LRP) which involves 2E-VRP with location decisions. We refer to Prodhon and Prins (2014) for a detailed review. The 2E-VRP-CO, however, is different in two ways from the 2E-LRP. First, we focus on operational cost minimization and therefore do not include fixed opening costs of locations. Second, every customer is visited exactly once in the 2E-LRP whereas a fraction of the customers do not have to be visited in the 2E-VRP-CO due to the covering locations.

Location routing problem where customers are serviced via two separate methods (e.g., home delivery and lockers) has been investigated in Stenger et al. (2012), Zhou et al. (2016) and Veenstra et al. (2018). The decision involves where to open lockers and how to route the visits to customers that are not covered by lockers. We extend the work by Stenger et al. (2012) and Zhou et al. (2016) by considering a two-echelon structure and heterogeneous vehicles per echelon. Although the work by Veenstra et al. (2018) considers distinct fleets to serve the customers (without two-echelon structure), the problem is more restrictive than ours as all customers within the covering radius of an opened covering location are excluded from the routing and cannot be served by cargo bikes. To the best of authors' knowledge, the only two-echelon problem examining covering locations is presented by Zhou et al. (2018). Their model considers multiple central warehouses, but there is no split delivery on the first echelon. Moreover, trucks visit satellites and then from satellites cargo bikes must visit covering locations on the second echelon. However, in our setting, trucks visit satellites and covering locations due to the large amount of parcels to be delivered, and thus their model differs significantly from ours.

The remainder of this paper is organized as follows. In Section 2, we formulate the 2E-VRP-CO and show that it generalizes the 2E-VRP and the SFL-VRP without duration constraints. In Section 3, we describe the ALNS heuristic. In Section 4, the results of the computational experiments are presented. Finally, we discuss in Section 5 the effect of different problem specific parameters and other managerial insights. Conclusions are provided in Section 6.

2. Problem formulation

The two-echelon vehicle routing problem with covering options (2E-VRP-CO) is defined on a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$. The set of vertices $\mathcal{V} = \{0\} \cup \mathcal{V}^{\mathcal{L}} \cup \mathcal{V}^{\mathcal{S}} \cup \mathcal{V}^{\mathcal{C}}$ contains the depot $\{0\}$, the set of covering locations $\mathcal{V}^{\mathcal{L}}$, the set of satellite locations $\mathcal{V}^{\mathcal{S}}$, and the customer vertices $\mathcal{V}^{\mathcal{C}}$. For all arcs $(i, j) \in \mathcal{A} := \{(i, j) \mid i, j \in \mathcal{V}, i \neq j\}$, the length and travel costs are denoted by d_{ij} and c_{ij}, respectively. We further define the set of arcs in the first echelon $\mathcal{A}^1 := \{(i, j) \in \mathcal{A} \mid i, j \notin \mathcal{V}^\mathcal{C}\}$, and the set of arcs in the second echelon $\mathcal{A}^2 := \mathcal{A} \setminus \mathcal{A}^1$.

On the first echelon, there are m^1 trucks with capacity Q^1 that travel from the depot to satellite locations and covering locations. On the second echelon, there are m^2 cargo bikes available with capacity Q^2 that can be assigned to any satellite location, provided that at most $\overline{m}_k$ cargo bikes can depart from satellite location $k \in \mathcal{V}^\mathcal{S}$. Trucks and cargo bikes can only be used on the first and second echelon, respectively. Finally, each vehicle (i.e., both trucks and cargo bikes) makes a single tour, starting and ending at the same location, and not all vehicles have to be used.

Each customer $i \in \mathcal{V}^\mathcal{C}$ has a demand $q_i > 0$ and is either visited by a single cargo bike or serviced through a single covering location. A covering location $j \in \mathcal{V}^\mathcal{L}$ can serve customers located within a radius r_j from its (geographical) location. The cost of connecting a customer $i \in \mathcal{V}^\mathcal{C}$ to covering location $j \in \mathcal{V}^\mathcal{L}$ equals ℓ_{ji}.

For the first echelon, let $x_{ij} \in \mathbb{N}_+$ and $w^1_{ij} \in \mathbb{R}$ be variables describing the number of trucks traversing and the amount of parcels being transported over $(i, j) \in \mathcal{A}^1$, respectively. For the second echelon, let $y_{ijk} \in \{0, 1\}$ being equal to 1 if edge $(i, j) \in \mathcal{A}^2$ is traversed with a cargo bike from satellite location $k \in \mathcal{V}^\mathcal{S}$, and let w^2_{ijk} denote the amount of parcels being transported along arc $(i, j) \in \mathcal{A}^2$. Consequently, the binary decision variable $v_k \in \{0, 1\}$ equals 1 if satellite or covering location $k \in \mathcal{V}^\mathcal{S} \cup \mathcal{V}^\mathcal{L}$ is used, and is 0 otherwise. Finally, we let $z_{ki} \in \{0, 1\}$ be equal to 1 if $i \in \mathcal{V}^\mathcal{C}$ is serviced through k, and 0 otherwise.

Mixed Integer Programming (MIP) formulations of 2E-VRP-CO can be derived differently, e.g., as in Veenstra et al. (2018) and Grangier et al. (2016). We experimented with these different compact formulations and found that the following 2E-VRP-CO formulation (P), partially based on Perboli et al. (2011) and Veenstra et al. (2018), performs better numerically.

$$\min \sum_{(i,j)\in\mathcal{A}^1} c_{ij}x_{ij} + \sum_{k\in\mathcal{V}^\mathcal{S}} \sum_{(i,j)\in\mathcal{A}^2} c_{ij}y_{ijk} + \sum_{k\in\mathcal{V}^\mathcal{L}} \sum_{j\in\mathcal{V}^\mathcal{C}} \ell_{kj}z_{kj} \tag{1}$$

$$\text{s.t.} \sum_{i\in\mathcal{V}^\mathcal{L}\cup\mathcal{V}^\mathcal{S}} x_{0i} \le m^1 \tag{2}$$

$$\sum_{k\in\mathcal{V}^\mathcal{S}} \sum_{i\in\mathcal{V}^\mathcal{C}} y_{kik} \le m^2 \tag{3}$$

$$\sum_{i:(i,j)\in\mathcal{A}^1} x_{ij} = \sum_{i:(j,i)\in\mathcal{A}^1} x_{ij}x_{ji} \quad \forall j \in \{0\} \cup \mathcal{V}^\mathcal{L} \cup \mathcal{V}^\mathcal{S} \tag{4}$$

$$\sum_{i:(j,i)\in\mathcal{A}^2} y_{jik} = \sum_{i:(i,j)\in\mathcal{A}^2} y_{ijk} = z_{kj} \quad \forall k \in \mathcal{V}^\mathcal{S}, j \in \mathcal{V}^\mathcal{C} \tag{5}$$

$$\sum_{i\in\mathcal{V}^\mathcal{C}} y_{kik} = \sum_{i\in\mathcal{V}^\mathcal{C}} y_{ikk} \le v_k\overline{m}_k \quad \forall k \in \mathcal{V}^\mathcal{S} \tag{6}$$

$$v_k \le \sum_{i:(k,i)\in\mathcal{A}^2} x_{ki} \quad \forall k \in \mathcal{V}^\mathcal{L} \cup \mathcal{V}^\mathcal{S} \tag{7}$$

$$\sum_{k\in\mathcal{V}^\mathcal{S}\cup\mathcal{V}^L} z_{ki} = 1 \quad \forall i \in \mathcal{V}^\mathcal{C} \tag{8}$$

$$d_{ki}z_{ki} \le r_k v_k \quad \forall k \in \mathcal{V}^\mathcal{L}, i \in \mathcal{V}^\mathcal{C} \tag{9}$$

$$\sum_{i\in V_C} z_{ki}q_i = \sum_{i:(i,k)\in\mathcal{A}^1} w^1_{ik} - \sum_{i:(k,i)\in\mathcal{A}^1} w^1_{ki} \quad \forall k \in \mathcal{V}^\mathcal{S} \cup \mathcal{V}^\mathcal{L} \tag{10}$$

$$\sum_{i\in\mathcal{V}^\mathcal{C}} q_i = \sum_{i:(0,i)\in\mathcal{A}^1} w^1_{0i} \tag{11}$$

$$0 \le w^1_{ij} \le Q^1 x_{ij} \quad \forall (i, j) \in \mathcal{A}^1 \tag{12}$$

$$\sum_{i\in\mathcal{V}^\mathcal{C}\cup\{k\}} w^2_{ijk} - \sum_{i\in\mathcal{V}^\mathcal{C}\cup\{k\}} w^2_{jik} = z_{kj}q_j \quad \forall k \in \mathcal{V}^\mathcal{S}, \; j \in \mathcal{V}^\mathcal{C} \tag{13}$$

$$\sum_{i\in\mathcal{V}^\mathcal{C}} w^2_{ikk} - \sum_{i\in\mathcal{V}^\mathcal{C}} w^2_{kik} = -\sum_{i\in V_C} z_{ki}q_i \quad \forall k \in \mathcal{V}^\mathcal{S} \tag{14}$$

$$0 \le w^2_{ijk} \le \left(Q^2 - q_i\right) y_{ijk} \quad \forall k \in \mathcal{V}^\mathcal{S}, (i, j) \in \mathcal{A}^2 \tag{15}$$

$$\sum_{i\in\mathcal{V}^\mathcal{L}\cup\mathcal{V}^\mathcal{S}} w^1_{i0} + \sum_{k\in\mathcal{V}^\mathcal{S}} \sum_{i\in\mathcal{V}^\mathcal{C}} w^2_{ikk} = 0 \tag{16}$$

$$y_{ijk} \in \{0, 1\} \quad \forall k \in \mathcal{V}^\mathcal{S}, (i, j) \in \mathcal{A}^1 \tag{17}$$

$$x_{ij} \in \mathbb{N}_+ \quad \forall (i, j) \in \mathcal{A}^1 \tag{18}$$

$$v_k \in \{0, 1\} \quad \forall k \in \mathcal{V}^\mathcal{S} \cup \mathcal{V}^\mathcal{L} \tag{19}$$

$$z_{ki} \in \{0, 1\} \quad \forall k \in \mathcal{V}^\mathcal{S} \cup \mathcal{V}^\mathcal{L}, i \in \mathcal{V}^\mathcal{C} \tag{20}$$

The Objective (1) minimizes the sum of the traveling costs and the connection costs. Constraints (2) and (3) restricts the total number of trucks and cargo bikes used. Constraints (4) and (5) are flow conservation constraints for the truck and cargo bikes respectively. Constraint (6) guarantees that cargo bikes can only depart from an opened satellite and have to return to the same satellite. Constraint (7) ensures that an intermediate location can only be opened if it is serviced by at least one truck. Constraint (8) assigns each customer to a satellite location or a covering location. Constraint (9) states that only customers within the covering range can be serviced by a covering location. Constraints (10)–(15) supervise the amount of parcels traversing over all the arcs. Additionally, these constraints prevent subtours on both echelons. Constraints (12) and (15) ensure that the capacity of vehicles is not exceeded. Constraint (16) ensures that all vehicles should be empty upon return.

The following additional valid inequalities from Perboli and Tadei (2010) can be used to strengthen the linear programming relaxation of (P).

$$\sum_{i,j\in\mathcal{V}^{\mathcal{C}'}\cup\{h\}} y_{ijk} \le \sum_{j\in\mathcal{V}^{\mathcal{C}'}} z_{kj} \qquad \forall k \in \mathcal{V}^\mathcal{S}, \; \forall h \in \mathcal{V}^\mathcal{C}, \; \mathcal{V}^{\mathcal{C}'} \subset \mathcal{V}^\mathcal{C}, \; \left|\mathcal{V}^{\mathcal{C}'}\right| = 2 \tag{21}$$

Note that the 2E-VRP-CO is equivalent to the 2E-VRP presented by Perboli et al. (2011) by taking $\mathcal{V}^\mathcal{L} = \varnothing$ in (P). Also, the SFL-VRP presented by Veenstra et al. (2018) can be derived from 2E-VRP-CO by taking $\mathcal{V}^\mathcal{S} = \{0\}$.

Proposition 1. *Consider the 2E-VRP-CO with $\mathcal{V}^\mathcal{S} = \{0\}$, $\ell_{ji} = 0$ $\forall j \in \mathcal{V}^\mathcal{L}, \forall i \in \mathcal{V}^\mathcal{C}$, and with travel cost $\tilde{c}_{ij} = c_{ij} + F_j$ if $j \in \mathcal{V}^\mathcal{C}$ and c_{ij} otherwise. Here, F_j are the fixed opening costs for a covering location as used by Veenstra et al. (2018). If we impose the additional constraints*

$$x_{ij} \le 1 \qquad \forall j \in \mathcal{V}^\mathcal{L}, \; \forall i \in \{0\} \cup \mathcal{V}^\mathcal{S} \cup \mathcal{V}^\mathcal{L} \tag{22}$$

$$r_j v_j \leq d_{ij} + \sum_{k \in \mathcal{V}^{\mathcal{L}}} z_{ki} M \qquad \forall j \in \mathcal{V}^{\mathcal{L}},\ \forall i \in \mathcal{V}^{\mathcal{C}} \tag{23}$$

the 2E-VRP-CO is equivalent to the SFL-VRP as introduced by Veenstra et al. (2018) without duration constraints.

Proof. Let an instance of the specific 2E-VRP-CO be given as described above. Then there is only one satellite location at the same location as the depot. Therefore, the first echelon trucks only service covering locations and the second echelon routes depart from the depot location. This is identical to respectively the locker and and patient routes for the SFL-VRP, as presented in Veenstra et al. (2018). The restrictions imposed by additional constraint (22) governs that at most a single truck is allowed to traverse on each first echelon arc and constraint (23) that each customers is served by a covering location if it is in range of an opened covering location. □

3. Adaptive large neighborhood search heuristic

Adaptive Large Neighborhood Search (ALNS) consists in iteratively improving solutions by applying *destroy* operators, which break down a part of the solution, and *repair* operators, which rebuild the destroyed solution in a sophisticated way. By continuously adapting the probabilities by which operators are selected, based upon their success, a large range of solutions of potential high quality will be explored.

The general outline of the ALNS procedure is described in Algorithm 1 . After initialization, destroy and repair operators are iteratively applied to create new best solutions until the stopping criterion is met. A new best solution is accepted based on a simulated annealing criterion. The advantage of this acceptance method is that local optima can be escaped, by accepting solutions which do not result in immediate improvements. Therefore, a larger part of the solution space can be examined which increases diversity of the search. If a new best solution is accepted, a local search procedure is started on that solution and operator probabilities are updated afterward.

Algorithm 1: ALNS Heuristic for the 2E-VRP-CO

begin ALNS heuristic
  $s_{\text{best}}, s^*, s \leftarrow$ InitialSolution
  $\pi \leftarrow$ InitializeOperatorProbabilities, $i_{\text{local}}, i_{\text{restart}} \leftarrow 0$
  while *StoppingCriterionNotReached* **do**
    $s' \leftarrow s$
    if $i_{local} < \omega_{grace}$ **then**
      $s' \leftarrow$ Destroy(s', D_S, π)
    else
      $s' \leftarrow$ Destroy(s', D_L, π)
      $i_{\text{local}} \leftarrow 0$
    $s' \leftarrow$ Repair(s', π)
    if *AcceptanceCriterion*(s', s) *or* $i_{local} = 0$ **then**
      $s \leftarrow$ LocalSearch(s')
      if $f(s) < f(s^*)$ *or* $i_{local} = 0$ **then**
        $s^* \leftarrow s,\ i_{\text{local}} \leftarrow 0$
      if $f(s) < f(s_{best})$ **then**
        $s_{\text{best}} \leftarrow s,\ i_{\text{restart}} \leftarrow 0$
    if $i_{restart} > \omega_{restart}$ **then**
      $s \leftarrow s_{\text{best}},\ i_{\text{restart}} \leftarrow 0$
    $\pi \leftarrow$ UpdateOperatorProbabilities(π)
    $i_{\text{local}} \leftarrow i_{\text{local}} + 1,\ i_{\text{restart}} \leftarrow i_{\text{restart}} + 1$
  return s_{best}

The ALNS heuristic is run in parallel on multiple threads. After ω_{restart} iterations, the four threads restart from the current best solution found by all threads. The efficiency of this approach on large-scale problems has recently been advocated by Schrotenboer et al. (2019).

Based on the ALNS heuristic of Hemmelmayr et al. (2012), a distinction is made between two types of destroy operators; D_L are the large destroy operators that change the configuration of available intermediate locations, and D_S are the small destroy operators that only effect a limited part of the solution. To ensure that a new configuration of opened intermediate locations can be thoroughly examined a grace period of ω_{grace} iterations is considered in which no large destroy operator can be employed.

In the following we discuss the destroy and repair operators, the acceptance criteria, the local search, and the procedures to update operator probabilities in detail.

3.1. Destroy operators

There are in total twelve destroy operators applied in the algorithm that removes part of the current solution. The destroy operators are divided into *large destroy operators*, that change the set of satellite locations and covering locations available, and *small destroy operators* that change the set of routes and their customers.

3.1.1. Small destroy operators

The in total six small destroy operators are divided into two *random* and four *guided small destroy operators*. The two *random small destroy operators* remove up to η customers, where η is a random integer within range $[\underline{q}, \bar{q}]$. The *random customer removal* operator selects and removes at random η customers from the solution. The *random route removal* operator selects at random a cargo bike route and removes all customers from that route until at least η customers are removed from the solution.

In Algorithm 2 , the general outline of the four *guided small destroy operators* is provided, each of which sort the customers according to an operator-specific metric from small to large. Each of the operators removes η customers from the solution, where selection is done based on the customer ranking and a measure of randomness $\rho \geq 1$ (Breunig et al., 2016; Hemmelmayr et al., 2012; Ropke and Pisinger, 2006). Customers are iteratively selected (and removed) by drawing indices $y = \lceil U(0,1)^p \times$ (nr customers in the ranking)$\rceil$. This ensures that for larger values of ρ, customers with a higher rank (i.e., a lower index) are more likely to be selected. For $\rho = 1$ the selection method is completely random, whereas for $\rho \to \infty$ the selection becomes almost surely deterministic as the η highest-ranked customers will be removed (Franceschetti et al., 2017).

In the following, we define the metrics used for each of the four guided small destroy operators included in the ALNS. The first guided small destroy operator is the *related removal* operator. It first selects a "seed" customer at random, and consequently, the metric that determines the ranking of customers equals the distance to the selected "seed" customer. Indeed, the selected "seed" customer has the lowest value and will therefore be first in the

Algorithm 2: Outline of the *small guided destroy operators*

begin Removing η customers based on a specific ranking
  Ranking all $|\mathcal{V}^{\mathcal{C}}|$ customers according to a metric.
  while *Number of customers selected* $< \eta$ **do**
    $y = \lceil U(0,1)^{\rho} \times$ (nr customers in the ranking)$\rceil$
    Remove customer at the y-th index of the ranking from the solution and ranking.

ranking. The *worst removal* operator uses as a metric the so-called customer removal gain. This is defined as the difference in objective value between the current solution and the solution without the customer.

The *minimum quantity removal* operator uses as metric the demand quantity of each customer. The idea behind this operator is that customers with low demand can be moved more easily to other routes in the solution. The fourth guided small destroy operator is the *least used vehicle removal*, which orders vehicle routes based on the total load of the vehicle. This operator thereby deviates from Algorithm 2 in the sense that cargo bikes are ranked instead of customers. Cargo bikes are then removed until at least η customers are removed from the solution, similar as described in Algorithm 2. Similar as described in Grangier et al. (2016) and the 'Remove single node routes' by Breunig et al. (2016), this operator aims to minimize the amount of vehicles used.

3.1.2. Large destroy operators

We consider six so-called *large destroy operators* that focus on beneficial configurations for the satellite and covering locations (Hemmelmayr et al., 2012). For both the covering locations and the satellite locations, we consider the *location close, location open*, and *location swap* operators. Since the operators work similar for satellite and covering locations, we explain the operators once in general terms with the general descriptor "intermediate location".

The *location close* operator selects a single random intermediate location, and removes all customers served by that location from the solution. The *location open* operator opens a random intermediate location from the set of currently closed locations. To intensify the search around newly opened intermediate locations, the η nearby customers are randomly removed from the current solution. This is is done by the related removal operator, where the opened location acts as the seed. The *location swap* closes a random intermediate location and opens another with the probability inversely related to the distance with the closed intermediate location. All customers served by the closed location are removed. Then, customers around the opened location are removed by the related removal operator, with the opened location as 'seed', until η customers are removed in total.

3.2. Repair operators

Customers are reinserted into the solution by the repair operators. The repair operators only consider feasible insertion options without the opening of additional satellite and covering locations. The overall framework of the repair phase is similar to Breunig et al. (2016). First all customers are coupled with one of the intermediate locations, and the second-echelon routes (or assignments) are constructed. Thereafter, based on the requested demand at the intermediate locations, the first echelon routes are reconstructed.

The ALNS includes three insertion operators. The first two operators are greedy insertion methods with a cheapest feasible insertion sub-procedure that inserts customers one by one at their cheapest location (see, e.g. Schrotenboer et al., 2018). Whereas a traditional cheapest feasible insertion method evaluates the insertion costs of all the customers after each customer insertion, the sub-procedure that we use only evaluates the insertion costs of a single customer after each customer insertion. The order in which the customers are inserted are based on an operator-specific ranking of customers.

The *Greedy Insertion* operator selects the customers to be inserted at random and inserts each customer at the position that minimizes the insertion cost. In other words, it uses the cheapest feasible insertion sub-procedure with a random ranking of customers. The *Greedy Insertion Perturbed* operator is similar but enforces additional randomness by multiplying the insertion costs with a perturbation factor, uniformly distributed on the interval $[1-\tau, 1+\tau]$.

For the 2E-VRP-CO, we introduce the *Greedy Insertion Regret* operator, which can be regarded as a combination of the common *Greedy Insertion* and *K-Regret Insertion* operators (Ropke and Pisinger, 2006). We rank the customers to be inserted by their distance to the nearest customer which is already routed. The customers who are located furthest away are ranked the highest. The customers within range of a covering location are always assigned the lowest priority, since they can always be serviced via that specific covering location. The customers are then inserted one-by-one in a similar fashion as how the the related removal operators work; Higher ranked customers have a higher probability to be inserted. This operator is different from the K-Regret Insertion operator as the insertion costs of all customers to insert is not recalculated each time a customer has been inserted, thereby reducing the computation time.

For all the repair operators, in case no feasible insertion location is available for a customer, the repair operator is restarted with an additionally, completely random removed customer. As the ranking of customers in the repair operators is non-deterministic, restarting a repair operator also implies a new ranking of customers.

Due to the inclusion of the Greedy Insertion Regret operator, we omit three commonly used operators from the ALNS. The *Random Insertion* operator, which positions customers in random order at random locations, did not provide good solutions for the 2E-VRP-CO. The *Basic Greedy* and the *K-Regret* operators, introduced by Ropke and Pisinger (2006), extend the computation time significantly without improvements in solution quality.

After the second echelon routes are reconstructed, we recreate the first echelon routes. At first, we determine if the current first echelon routes are still feasible. If not feasible, we completely reconstruct the first echelon routes, since the changes in the second echelon routes typically have a substantial effect on the required demand at the intermediate locations. For any intermediate location with a requested demand larger than the truck capacity, we assign dedicated full truckload routes to that intermediate location until the remaining demand is smaller than a full truckload, as is done by Breunig et al. (2016) and Wang et al. (2017). Routes for the remaining intermediate locations are created using the *Greedy Insertion* operator.

We create 100 initial solutions of the ALNS (the input of Algorithm 1) by applying the *Greedy Insertion* operator with all customers are present in the customer pool. Each thread selects at random one of the 10 best found initial solutions to start the ALNS.

3.3. Acceptance criteria

The acceptance criterion defines whether a new solutions s' is accepted as the current solution s (see Algorithm 1). If s' has a lower objective value than s, or if it is the result of a large destroy operator, we replace s with s'. If the solution s' is the result of a small destroy operator, we impose a simulated annealing criterion (Kirkpatrick et al., 1983) in order to determine the probability of accepting the solution. The probability of acceptance is equal to $e^{-(f(s')-f(s))/T}$, where $T > 0$ is the temperature at the given iteration. At the start of the ALNS heuristic the temperature is initialized at $T = T_{\text{start}}$. At each iteration the temperature multiplied by a factor $\kappa \in [0, 1]$, denoted as the cooling rate.

3.4. Local search

If a new solution is accepted, we perform a local search to intensify the search. For both echelons the same local search operators are used sequentially, namely: *1-0-Exchange, Intra-Swap, Intra-2-Opt, Inter-Swap*, and *Inter-2-Opt*. These operators are performed either within (intra) or between (inter) routes, and they evaluate swapping customers (*swap*), swapping edges (*2-opt*) and reinserting customers (*1-0-Exchange*), see e.g. Veenstra et al. (2018). Each operator is run in a first-improvement fashion, i.e., directly applying a move when it reduces overall costs, until no new improvements can be found. Then, the next operator is performed until all operators are considered sequentially. The local search on the second echelon routes is performed for each satellite location separately, so that the requested demand remains unchanged at the intermediate locations. Afterward, the five local search operators are performed on the truck routes.

3.5. Updating operator probabilities

To select the destroy and repair operators, we use a roulette wheel selection principle with relative weights assigned to all operators, as proposed by Pisinger and Ropke (2007). This selection method adjusts the operator weights based on their performance over the previous iterations. The accumulated performance score for operator i, denoted by ψ_i, is increased by the coefficients σ_1, σ_2 or σ_3 respectively in case a new global best solution is found, the current solution is improved while the global best solution remains unchanged or the solution is accepted without improving the objective. Over a segment consisting of 100 iterations, the weight θ_i of operator i is updated as in:

$$\theta_i \leftarrow (1-\lambda)\theta_i + \lambda \frac{\psi_i}{n_i} \tag{24}$$

where n_i is the number of times an operator is selected and λ is the reaction factor, which controls how fast the weight adjustment procedure reacts to changes in the scores.

4. Computational experiments

In this section, we assess the performance of our ALNS by solving well-known benchmark instances for the 2E-VRP and the SFL-VRP, as well as solving newly created 2E-VRP-CO instances. For the 2E-VRP-CO instances, the ALNS performance is compared with solving the MIP formulation as described in Section 2 with CPLEX 12.8. The ALNS heuristic is programmed in *C++*, and at most four parallel running threads are deployed. All experiments are deployed on a Intel Xeon E5 2680v3 runnig at 2.5 GHz.

4.1. Description of the benchmark sets

As the 2E-VRP and the SFL-VRP are special cases of the 2E-VRP-CO, we benchmark our ALNS heuristic against the best-known solutions on established benchmark instances for these problems. For the 2E-VRP, we consider three well-known sets of benchmark instances (see, e.g., Breunig et al., 2016; Hemmelmayr et al., 2012), namely ‘Set 2’, ‘Set 3’, and ‘Set 5’[1]. For the SFL-VRP, we consider the 72 instances provided by Veenstra et al. (2018).

For the 2E-VRP-CO, we construct two new sets of instances by expanding the ‘Set 2’ and ‘Set 3’ 2E-VRP instances. For each instance we introduce a number of covering locations equal to the amount of satellite locations. Each covering location is positioned at the same location as a randomly selected customer, with the restraint that the customer is not within the covering range of another covering location. In case there is no feasible location left to place a covering location, we place the remaining covering locations at random positions on the customer plane. The customer plane is the smallest square, with the edges parallel to the axes, that contains all customers. The covering range equals 25% of diagonal customer plane length for each covering location, and is called d_{diag}. The connection cost, incurred if customer i is serviced by covering location j, equals $\ell_{ji} = \alpha d_{ji} + \beta d_{\text{diag}}$, where α is the distance dependent connection cost factor and β a fixed connection cost factor. For all instances we set $\alpha = \beta = 0.25$. In Section 5, we vary this parameter and see how it effects the resulting solutions. In total 108 instances are created, 54 instances for parameter tuning and model validation.

4.2. ALNS heuristic parameter tuning

We performed a preliminary computational campaign to tune the parameters of the ALNS heuristic. The parameters are tuned on 54 new 2E-VRP-CO instances, created in same manner described in Section 4.1.

We started the parameter tuning by considering the parameter settings described by Hemmelmayr et al. (2012). Subsequently, we iteratively tested a range of values for each parameter, only accepting improvements. This is similar to the approach taken by, for instance, Veenstra et al. (2018).

The final values of the heuristic parameters are as follows. At the start of the heuristic 75% of the satellite locations and 30% of the covering locations are opened at random. At each iteration a random amount of customers between $\underline{q} = \max(0.2|\mathcal{V}^{\mathcal{C}}|, 5)$ and $\overline{q} = \min(0.4|\mathcal{V}^{\mathcal{C}}|, 40)$ are removed from the current solution. The grace period, ω_{grace}, is equal to 220 iterations. The measure of randomness for the destroy operators is $\rho = 3$. The perturbation parameter τ used in the repair phase is 0.2. For the simulated annealing based acceptance criterion, the starting temperature is selected such that a solution 5% worse than the initial solution is accepted with a 20% probability. At each successive iteration the cooling rate decreases with $\kappa = 0.9999$. The probabilities of selecting the destroy and repair operators are governed by a roulette wheel selection mechanism. The values of the control parameters for this mechanism are after extensive tests set to 60, 30, 20 and 0.55 for σ_1, σ_2, σ_3, and λ respectively. The algorithm restarts from the current best solution after 10,000 iterations without a new global improvement. At that moment the operator selection probabilities are also reset. The algorithm ends after one million iterations.

To solve the SFL-VRP we make minimal adjustments to the ALNS heuristic. Any change to a solution that violates the duration constraints is not allowed. Also, operators not relevant for SFL-VRP settings are removed, including *Satellite Removal, Satellite Opening, Satellite Swap, Minimum Quantity Removal, Random Route Removal* and *Least Used Vehicle Removal*. Furthermore, the grace period length is reduced to 10 iterations and restarts are executed after 100 iterations without finding a new global improvement. We find that a smaller grace period provides a better trade-off between exploring more configurations of opened and closed covering locations and performing a stronger examination of each configuration.

4.3. ALNS performance on SFL-VRP and 2E-VRP instances

The results of solving the benchmark instances of the SFL-VRP and 2E-VRP are in summarized form presented in Tables 1 and 2, respectively. Results on an individual instance level can be found in Appendix Appendix A. We grouped the individual SFL-VRP instances based on the number of customers, resulting into 8 sets of

[1] https://www.univie.ac.at/prolog/research/TwoEVRP

Table 1
Averaged results for the SFL-VRP benchmark instances of Veenstra et al. (2018).

Set	$\lvert\mathcal{V}^C\rvert$	$\lvert\mathcal{V}^L\rvert$	Veenstra et al. (2018)			BKS	ALNS heuristic					
			Avg. 5	Best	$t(s)$		Avg. 5	Best	Gap Veenstra (%)	Gap BKS (%)	$t(s)$	$t^*(s)$
1	30	10–50	2503.5	2503.4	14	2503.4	2503.7	2503.4	0.00	0.00	96	3
2	40	10–50	3719.0	3719.0	24	3719.0	3719.8	3719.0	0.00	0.00	108	2
3	50	10–50	3996.4	3996.3	36	3996.3	3997.0	3996.3	0.00	0.00	152	11
4	60	10–50	4030.2	4022.4	52	4014.1	4014.1	4014.1	−0.13	0.00	148	7
5	70	10–50	4163.0	4162.9	83	4162.9	4165.1	4162.9	0.00	0.00	133	5
6	80	10–50	4686.0	4686.0	105	4684.9	4685.1	4684.9	−0.03	0.00	179	12
7	90	10–50	4596.4	4596.4	135	4596.4	4596.2	4595.3	−0.04	−0.04	160	12
8	100	10–50	4703.1	4702.6	179	4702.6	4704.9	4702.6	0.00	0.00	198	14
Avg.			*4049.7*	*4048.6*	*78*	*4047.5*	*4048.2*	*4047.3*	*−0.03*	*−0.01*	*147*	8

Table 2
Average statistics on the 2E-VRP benchmark instances of Breunig et al. (2016). Here m^1 and m^2 denote the number of trucks and cargo bikes, respectively.

Instance Set	$\lvert\mathcal{V}^C\rvert$	$\lvert\mathcal{V}^S\rvert$	m^1	m^2	BKS	Avg. 5	Best	Gap BKS (%)	$t(s)$	$t^*(s)$
Set 2	21–50	2–4	3–4	4–5	578.27	578.30	578.27	0.00	135	1
Set 3	21–50	2–4	3–4	4–5	641.44	641.52	641.44	0.00	124	6
Set 5[a]	100–200	5–10	5	15–63	1118.81	1144.08	1131.47	0.98	770	417
Avg.					734.46	740.82	737.63	0.25	290	107

[a] We found two new best-known solutions; Instance ‘200-10-1’ improving from 1556.79 to 1553.75 and Instance ‘200-10-2b’ improving from 1002.85 to 1002.63.

Table 3
Comparison of a single and multi-threaded implementation of the ALNS.

	Multi-threaded				Single-threaded				Δ Avg. 5 (%)
	Avg. 5	Best	$t(s)$	$t^*(s)$	Avg. 5	Best	$t^*(s)$	$t(s)$	
SFL-VRP									
1	2503.7	2503.4	96	3	2505.1	2503.4	43	7	0.1
2	3719.8	3719.0	108	2	3720.5	3719.0	49	5	0.0
3	3997.0	3996.3	152	11	3996.6	3996.3	59	6	0.0
4	4014.1	4014.1	148	7	4014.2	4014.1	71	4	0.0
5	4165.1	4162.9	133	5	4165.4	4162.9	70	13	0.0
6	4685.1	4684.9	179	12	4686.2	4684.9	85	12	0.0
7	4596.2	4595.3	160	12	4597.4	4595.3	90	15	0.0
8	4704.9	4702.6	198	14	4711.2	4710.4	98	10	0.1
Avg.	*4048.2*	*4047.3*	*147*	*8*	*4049.6*	*4048.3*	*70*	*9*	*0.0*
2E-VRP									
2	578.30	578.27	135	1	579.54	578.62	54	6	0.2
3	641.52	641.44	124	6	643.59	642.03	49	6	0.3
5	1144.08	1131.47	770	417	1151.05	1140.96	236	155	0.6
Avg.	*740.82*	*737.63*	*290*	*107*	*743.77*	*740.35*	*98*	*43*	*0.4*

10 instances. For the 2E-VRP instances, we summarized the results based on the sets the instances belong to.

We compared the performance of the ALNS to the best-known solutions (BKS), which are marked with an asterisk if proven to be optimal. For the ALNS performance, we report the best and average (over 5 runs) objective, the gap in percentages with the best-known solution in the literature, the computation time, $t(s)$, and the average time until the optimal solution is found, $t^*(s)$. For the SFL-VRP instances, we additionally provide the difference with, and performance of Veenstra et al. (2018).

Regarding the SFL-VRP instances, see Table 1, it is observed that the ALNS provides high-quality, and often optimal, solutions. We find, or improve, all best-known solutions, resulting in a slight improvement of 0.03% over the heuristic of Veenstra et al. (2018). Compared to their heuristic, we find improvements on instances R028, R050 and R062 (see Appendix A, Table 9). In addition, we find a single new best-known solution (Instance R062). Regarding computation times, it is observed that the best solution is on average found in 8 seconds, with a total average runtime of 147 seconds. This is comparable to the run-times reported by Veenstra et al. (2018) who performed their experiments on a slightly faster CPU (Xeon Processor X5650 2.66 GHz).

Moving a step further away from the problem setting considered in this paper, we arrive at the 2E-VRP instances, see Table 2. Similar, high-quality, results as for the SFL-VRP instances are reported. Namely, the instances of Set 2 and 3 are all solved to their best-known solution, and the large instances can be solved to an average gap to the best-known solutions of 0.98%. During the execution of our algorithm, we discovered two new best-known solutions: For instance 200-10-1, we improve the objective from 1556.79 to 1553.75, and for instance 200-10-2b we improve the objective from 1002.85 to 1002.63.

We further investigate the performance of a single-thread implementation of the ALNS, and compare this with the multi-threaded implementation. The results are presented in Table 3. It is noticeable that the single-threaded implementation has smaller computation times then the multi-threaded implemen-

Table 4
Computational results for the 2E-VRP-CO instances ($\alpha = 0.25$, $\beta = 0.25$).

Instance	Results MIP								
	$\bar{z}$	$\underline{z}$	Gap (%)	t (s)	Avg. 5	Best	Gap MIP (%)	$t(s)$	$t^{*}(s)$
Set 2a									
E-n22-k4-s6-17	415.39	415.39	0.00	4	415.39	415.39	0.00	60	0
E-n22-k4-s8-14	362.38	362.38	0.00	3	362.38	362.38	0.00	60	0
E-n22-k4-s9-19	467.52	467.52	0.00	32	467.52	467.52	0.00	106	0
E-n22-k4-s10-14	330.88	330.88	0.00	1	330.88	330.88	0.00	98	0
E-n22-k4-s11-12	427.22	427.22	0.00	52	427.22	427.22	0.00	62	0
E-n22-k4-s12-16	386.56	386.56	0.00	17	386.57	386.57	0.00	66	0
E-n33-k4-s1-9	711.88	711.88	0.00	79	711.89	711.89	0.00	135	3
E-n33-k4-s2-13	714.63	714.63	0.00	155	714.63	714.63	0.00	95	1
E-n33-k4-s3-17	707.48	707.48	0.00	694	707.48	707.48	0.00	121	0
E-n33-k4-s4-5	687.33	687.33	0.00	76	687.33	687.33	0.00	82	0
E-n33-k4-s7-25	720.62	720.62	0.00	207	720.62	720.62	0.00	88	1
E-n33-k4-s14-22	760.88	760.88	0.00	79	760.88	760.88	0.00	91	0
Set 2b									
E-n51-k5-s2-4-17-46	572.03	514.88	11.10	62	530.76	530.76	0.00	151	3
E-n51-k5-s2-17	585.24	561.92	4.15	59	582.01	582.01	−0.55	118	8
E−n51-k5-s4-46	530.76	520.69	1.93	62	530.76	530.76	0.00	201	1
E-n51-k5-s6-12	558.95	530.02	5.46	63	554.81	554.81	0.00	120	2
E-n51-k5-s6-12-32-37	542.78	508.88	6.66	62	531.92	531.92	0.00	181	3
E-n51-k5-s11-19	583.04	560.86	3.95	60	581.64	581.64	0.00	155	6
E-n51-k5-s11-19-27-47	532.19	506.13	5.15	61	527.63	527.63	0.00	151	3
E-n51-k5-s27-47	536.30	513.28	4.48	63	532.39	532.39	−0.73	108	7
E-n51-k5-s32-37	552.28	530.20	4.16	65	552.28	552.28	0.00	116	1
Set 2c									
E-n51-k5-s2-4-17-46	592.95	547.26	8.35	56	571.78	571.78	−3.57	186	31
E-n51-k5-s2-17	590.66	555.71	6.29	61	580.84	580.84	−1.66	113	15
E-n51-k5-s4-46	666.82	627.37	6.29	59	665.89	665.77	−0.16	146	31
E-n51-k5-s6-12	574.84	538.98	6.65	63	567.42	567.42	0.00	124	5
E-n51-k5-s6-12-32-37	553.48	530.46	4.34	61	553.48	553.48	0.00	114	9
E-n51-k5-s11-19	616.34	572.40	7.68	60	602.44	602.44	-2.26	104	10
E-n51-k5-s11-19-27-47	579.32	514.75	12.54	58	530.76	530.76	0.00	207	4
E-n51-k5-s27-47	530.76	527.19	0.68	61	530.76	530.76	0.00	115	1
E-n51-k5-s32-37	712.77	666.33	6.97	55	703.32	703.32	-1.33	105	4
Set 3a									
E-n22-k4-s13-14	492.22	492.22	0.00	15	492.22	492.22	0.00	62	0
E-n22-k4-s13-16	481.40	481.40	0.00	3	481.40	481.40	0.00	65	0
E-n22-k4-s13-17	496.38	496.38	0.00	26	496.38	496.38	0.00	66	0
E-n22-k4-s14-19	454.33	454.33	0.00	30	454.33	454.33	0.00	60	0
E-n22-k4-s17-19	512.80	512.80	0.00	394	512.81	512.81	0.00	56	0
E-n22-k4-s19-21	501.28	501.28	0.00	213	501.28	501.28	0.00	69	0
E-n33-k4-s16-22	628.21	628.21	0.00	492	628.21	628.21	0.00	72	0
E-n33-k4-s16-24	655.85	613.84	6.84	33	651.46	651.46	-0.67	69	1
E-n33-k4-s19-26	664.91	651.71	2.03	39	664.91	664.91	0.00	82	5
E-n33-k4-s22-26	641.02	629.47	1.84	34	641.02	641.02	0.00	82	1
E-n33-k4-s24-28	670.43	637.26	5.20	36	670.43	670.43	0.00	121	1
E-n33-k4-s25-28	645.56	628.18	2.77	37	645.56	645.56	0.00	71	0
Set 3b & 3c									
E-n51-k5-s12-18	722.60	655.50	10.24	63	690.59	690.59	0.00	118	16
E-n51-k5-s12-41	698.06	639.00	9.24	59	683.05	683.05	0.00	168	21
E-n51-k5-s12-43	734.79	690.61	6.40	61	710.41	710.41	0.00	108	1
E-n51-k5-s13-19	604.45	536.99	12.56	63	560.73	560.73	0.00	186	3
E-n51-k5-s13-42	571.70	547.14	4.49	63	564.45	564.45	0.00	106	1
E-n51-k5-s13-44	572.26	549.63	4.12	58	564.45	564.45	0.00	113	1
E-n51-k5-s39-41	726.05	677.43	7.18	52	717.46	717.46	-1.18	104	2
E-n51-k5-s40-41	728.15	669.86	8.70	51	714.24	714.24	-1.31	140	30
E-n51-k5-s40-42	707.89	657.59	7.65	55	700.13	700.13	-1.10	206	29
E-n51-k5-s40-43	739.76	695.66	6.34	55	729.74	729.74	-1.35	105	1
E-n51-k5-s41-42	704.46	686.98	2.54	54	703.86	703.86	-0.09	104	2
E-n51-k5-s41-44	734.70	690.45	6.41	55	723.42	723.42	-1.54	184	19
Avg.	*596.18*	*573.04*	*3.91*	*84*	*589.37*	*589.37*	*-0.32*	*113*	*5*

tation. This is because the multiple threads apply different operators on different solution, which desynchronizes the computation times between the threads. The performance on the SFL-VRP instances is similar. However, the multi-threaded implementation is clearly more robust on the 2E-VRP instances, as it is on average 0.4% better than the single-threaded implementation. Furthermore, for the SFL-VRP instances, the running times of the single-thread implementation (70 s) and the running times of Veenstra et al. (2018) (78 s) are comparable.

4.4. Results on the 2E-VRP-CO instances

We provide computational results for the newly created 2E-VRP-CO instances in Table 4. In order to assess the solution quality of the ALNS heuristic, we tested against the results obtained by solving the MIP (Section 2) with CPLEX 12.8 allowing at most 3600

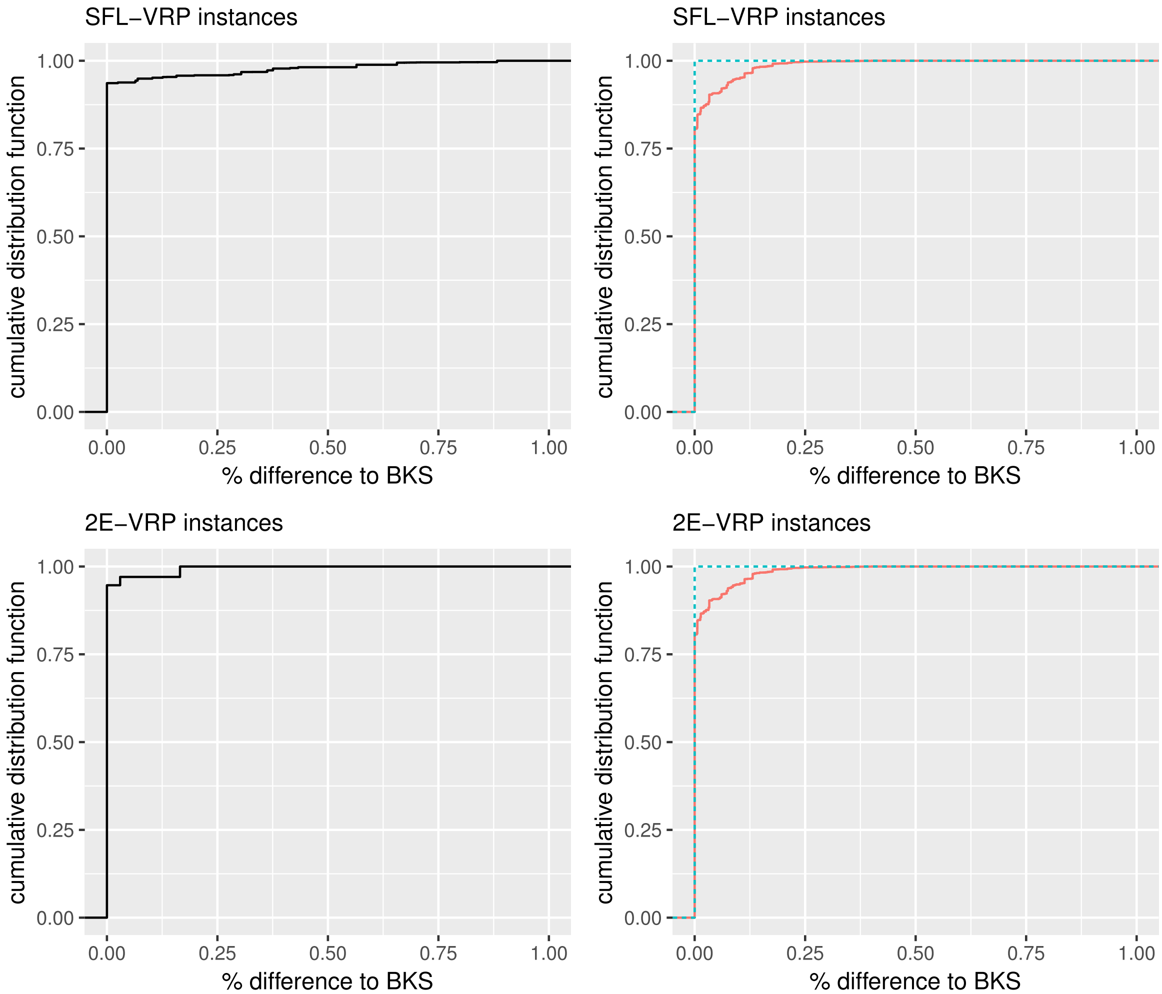


Fig. 2. Performance robusteness of the ALNS on SFL-VRP and 2E-VRP instances with $n \leq 100$. The left graphs show the empirical cumulative distribution function distribution of 100 runs of the ALNS. The right graphs show the empirical cumulative distribution functions of the minimum (dotted blue) and average (solid red) objective obtained with 5 runs.

seconds for solving. The instances involve less than 100 customers due to the computational limitations of the MIP formulation.

The instance names are similar as used in Breunig et al. (2016), e.g., instance E-n22-k4-s6-17 denotes an instance with 22 nodes, 4 cargo bikes and nodes 6 and 17 are satellite locations. For each instance, we report in addition to the ALNS performance, the upper bound $\bar{z}$, the lower bound, $\underline{z}$, the corresponding optimality gap, and the total computation time $t(s)$ resulting from solving the MIP.

From Table 4, we observe that for all the instances solved to optimality with the MIP, the ALNS finds the optimal solution as well. If the MIP could not be solved to optimality, the ALNS finds or improves the best found solution with on average 1.09%. The ALNS finds its best solution in on average 4 seconds with a total runtime of on average 100 seconds, whereas the MIP has an average runtime of 2381 seconds.

4.5. Impact of different ALNS elements

To investigate which elements of the ALNS contribute to its performance, we performed an additional round of experiments using the 2E-VRP, SFL-VRP, and 2E-VRP-CO instances. First, we analyze the ALNS runs leading to the results described in the previous section, and we count the average number of times (per instance) that applying an operator resulted into a new best solution. Second, we resolved all instances by removing a single operator, and looked to the solution degradation of the resulting heuristic.

We provide the results of these experiments in Table 5. Notably, although the large destroy operators (removal, opening and swapping intermediate locations) are predominantly included for diversification purposes, applying them still resulted into new best solutions. In addition, all operators seem to be able to move to new solutions, especially for the SFL-VRP and 2E-VRP-CO.

Regarding the solution degradation after removing a single operator, it is observes that especially for the 2E-VRP instances, it matters for each individual operator. An immediate solution degradation is observed if the covering location operators are excluded from the ALNS, which was to be expected as they form a crucial part of the 2E-VRP-CO. Finally, it is notable that solutions for the 2E-VRP degrade if the satellite location operators are excluded, indicating that in our ALNS these operators have an important role in the diversification of the search.

4.6. Robustness of the ALNS

It is important that the outcomes of the ALNS are robust so that reliable solutions are provided in practical scenarios. To investigate the robustness, we solve each of the considered instances 100 times and stored the objectives found by the ALNS. We use these to create empirical distribution functions of the observed difference to the best known solution. In addition, using the 100 runs of the ALNS on each instance we investigated the robustness of taking the average and minimum results of 5 ALNS runs by taking

Table 5
Sensitivity analysis and contribution of individual destroy and repair operators.

Operator	Average number of best solutions found by each operator per set			Solution degredation without the operator per set (%)		
	2E-VRP	SFL-VRP	2E-VRP-CO	2E-VRP	SFL-VRP	2E-VRP-CO
Covering Removal	–	14.56	0.35	–	4.66	0.07
Covering Opening	–	23.40	0.31	–	0.34	0.19
Covering Swap	–	20.18	0.17	–	0.01	0.00
Satellite Removal	0.07	–	0.11	−0.01	–	1.31
Satellite Opening	0.00	–	0.07	0.35	–	0.10
Satellite Swap	0.00	–	0.35	0.06	–	0.00
Random Related	33.35	48.88	23.37	0.04	−0.01	0.00
Related Removal	23.75	68.08	21.91	0.05	0.03	0.00
Worst Removal	32.79	60.51	26.26	0.04	0.00	0.00
Minimum Quantity Removal	25.93	–	16.44	0.14	–	0.00
Random Route Removal	7.71	–	6.02	0.01	–	0.00
Least Used Vehicle Removal	21.81	–	9.69	0.01	–	0.00
Greedy Insertion	55.06	78.75	35.43	0.07	0.00	0.00
Greedy Insertion Perturbed	46.42	73.14	29.61	0.07	0.00	0.00
Greedy Insertion Regret	55.99	83.72	52.91	0.08	0.00	0.00
Local Search (1st echelon)				0.08	0.00	0.00
Local Search (2nd echelon)				0.13	0.00	0.00

10,000 random samples of size 5 from the ALNS. The results are presented in Fig. 2.

Fig. 2 shows that all the observed differences with the best known solutions (BKS) are within 1%. The minimum over 5 runs always contains the BKS, as is showed by the dotted lines in the top and bottom right graphs. Also the mean performance of the ALNS is robust, since in more than 75% the average performance over 5 runs is equal to the BKS. This average is also within 0.25% of the BKS for all the 10,000 samples. Note that the 2E-VRP-CO instance are not presented in Fig. 2, since the outcomes were so robust that no noticeable differences to the best known solutions are found.

5. Managerial insights

In the following, we study the effects of employing covering locations within last mile delivery. As a base-case, we use the 72 newly created instances with $\alpha = \beta = 0.25$ as described in Section 4.1. By varying the parameters governing the connection costs, driving costs and covering range we will infer how, and when, covering locations are effective within the last-mile delivery. Recall that cargo-bikes are used as a general descriptor for easily manoeuvrable and environmental-friendly vehicles suitable for transportation in inner cities. The results presented in the following are, therefore, also generally applicable for other innovative forms of inner-city transportation, for instance the use of electric powered city freighters.

For each instance and configuration of parameters, we use the best solution over 5 ALNS heuristic runs. The results are shown in Figs. 3(a)–5(b), where the average length of the truck routes and cargo bike routes are depicted, as well as the amount of trucks, cargo bikes, satellite locations and covering locations used. Moreover, the percentage change in the total operational costs and the percentage of customers serviced by the covering locations (within brackets) are presented above the bars.

We observe that lower connection costs (Appendix A, Figs. 4(a)–4(b)), larger covering ranges (Appendix A, Fig. 5(a)) and higher penalties on cargo bikes (Appendix A, Fig. 5(b)) result in a higher usage rate of intermediate locations. By comparing two extreme scenarios, where the connection costs are respectively zero and infinite, we find that the introduction of covering locations can reduce the operational costs by 35.4%. Furthermore, the distance driven by cargo bikes can be reduced by 60.4%. On the other hand, since more intermediate locations have to be visited, the total length of the truck routes increases, with up to 37.8%.

Furthermore, we conclude that covering location are more beneficial when the covering range is larger. This emphasizes the importance of placing covering location at positions which are easily accessible. However, the marginal gain of extending the range decreases, due to the distance dependent connection costs and potential overlapping working areas of multiple covering locations.

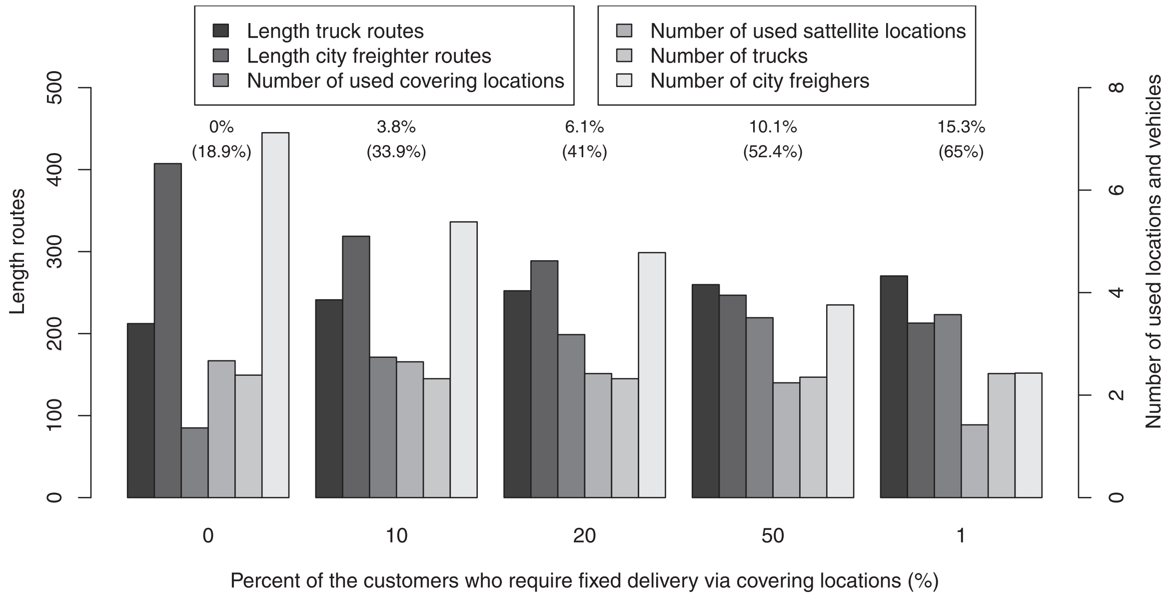
5.1. Customers with a preferred delivery method

An implicit assumption made in literature on combining routing with covering options, is that all customers can be serviced directly or via nearby covering locations, and that the decision is made by the distributor. However, some customers may have strong preferences for a particular delivery method. We, therefore, investigate the effect of fixing the delivery methods for different percentages of the customers on the resulting solutions.

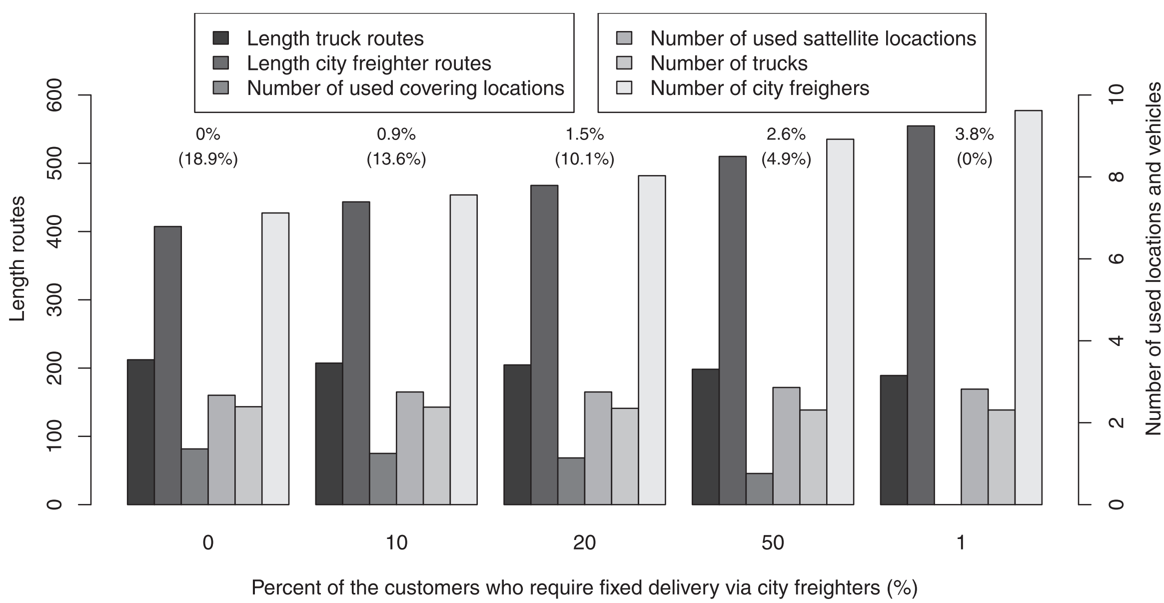
Fig. 3 (a) presents the effect of fixing a percentage of the customers to be serviced by covering locations and Fig. 3(b) depicts the effect when a percentage of the customers have to be serviced by cargo bikes. These situations are modelled by respectively setting the driving costs for cargo bikes or the connection costs to these customers equal to infinity. The customers with a predefined delivery method are selected at random. We assumed that distributors can still select through which covering locations these customers are served, when a customer is within range of multiple covering locations.

Even when only small groups of customers have a fixed delivery method we observe large effects. In case a fraction of the customers have to be serviced via covering locations, we observe that the percentage of customers reached via covering locations increases even further. Due to the forced opening of some of the covering locations, delivery via one of the opened covering locations becomes more advantageous for other customers as well.

On the other hand when a fraction of the customers have to be served via cargo bikes we observe the opposite. Less of the covering locations are employed and more of the customers are serviced directly. This is due to the fact that it is typically not profitable to open a covering location when a cargo bike already serves customers within the vicinity. Furthermore, we observe that the combined distance driven by the two types of vehicles and the overall operational costs are lowest when all customers can be served by both methods. This is advantageous from a cost and environmental



(a) The effect of fixed delivery via covering locations.



(b) The effect of fixed delivery via cargo bikes (also called city freighters.)

Fig. 3. Solution characteristics for customers with preferred delivery method.

perspective for the distributor and the customers. From these findings we can conclude that a good routing solution aims to serve customers near each other in a similar way. In practice this results in interesting situations where customers in a neighborhood should be rewarded if their delivery preferences are similar.

6. Conclusion

In this paper, we introduced the two-echelon vehicle routing problem with covering options (2E-VRP-CO). This problem arises when customers can be serviced through two types of intermediate locations. Delivery can occur via covering locations, from where customers within a specified covering range can collect the parcels by themselves, or via satellite locations, from where cargo bikes can service the customers directly at their home.

We develop an efficient ALNS heuristic for the 2E-VRP-CO, and we assess its performance on two special cases of the 2E-VRP-CO, the 2E-VRP and SFL-VRP. It appears that the ALNS is able to find high quality, and often optimal, solutions to these benchmark instances. In addition, we report for each of the two special cases a new best-known solution. On newly developed instances for the 2E-VRP-CO, we show that the ALNS provide superiour results (in terms of upper bounds) compared to a compact MIP formulation solved with CPLEX 12.8.

Moreover, managerial insights on the effects of introducing covering locations are presented for different configurations. Results show that covering locations can reduce the total operational costs with up to 35.4% depending on the connection cost structure.

We also examine a case where a fraction of customers requires a fixed method of delivery. This case leads to an increase in operational costs and the total distance driven, thus it is desirable to reward customers who provide the flexibility of multiple delivery options.

Interesting extensions on this research consist of examining the effects of incorporating return deliveries and time synchronization at the intermediate locations. Both of these expansions may increase the profitability of covering locations even further.

CRediT authorship contribution statement

David L.J.U. Enthoven: Conceptualization, Methodology, Software, Writing - original draft, Writing - review & editing. **Bolor Jargalsaikhan:** Conceptualization, Methodology, Formal analysis, Writing - original draft, Writing - review & editing, Supervision. **Kees Jan Roodbergen:** Conceptualization, Methodology, Formal analysis, Writing - original draft, Writing - review & editing, Supervision, Funding acquisition. **Michiel A.J. uit het Broek:** Conceptualization, Methodology, Formal analysis, Writing - original draft, Writing - review & editing, Supervision. **Albert H. Schrotenboer:** Conceptualization, Methodology, Formal analysis, Writing - original draft, Writing - review & editing, Supervision.

Acknowledgment

This work was partly supported by the Netherlands Organisation for Scientific Research (NWO) through grants no 438-13-216 (Michiel A. J. uit het Broek and Albert H. Schrotenboer) and no 439-16-612 (Kees Jan Roodbergen and Bolor Jargalsaikhan).

Appendix A. Detailed computational results

Table 6
Results part I/II for the 2E-VRP benchmark instances of Breunig et al. (2016).

Instance	$\lvert\mathcal{V}^c\rvert$	$\lvert\mathcal{V}^S\rvert$	m^1	m^2	BKS	ALNS 4-threads					ALNS single-threaded				
						Avg. 5	Best	Gap BKS (%)	$t(s)$	$t^*(s)$	Avg. 5	Best	Gap BKS (%)	$t(s)$	$t^*(s)$
Set 2a															
E-n22-k4-s6-17	21	2	3	4	417.07*	417.07	417.07	0.00	75	0	417.07	417.07	0.00	25	0
E-n22-k4-s8-14	21	2	3	4	384.96*	384.96	384.96	0.00	103	0	384.96	384.96	0.00	27	0
E-n22-k4-s9-19	21	2	3	4	470.60*	470.60	470.60	0.00	64	0	470.60	470.60	0.00	24	0
E-n22-k4-s10-14	21	2	3	4	371.50*	371.50	371.50	0.00	75	0	371.50	371.50	0.00	24	0
E-n22-k4-s11-12	21	2	3	4	427.22*	427.22	427.22	0.00	138	0	427.22	427.22	0.00	24	0
E-n22-k4-s12-16	21	2	3	4	392.78*	392.78	392.78	0.00	67	0	392.78	392.78	0.00	27	0
E-n33-k4-s1-9	32	2	3	4	730.16*	730.16	730.16	0.00	157	1	730.16	730.16	0.00	44	3
E-n33-k4-s2-13	32	2	3	4	714.63*	714.63	714.63	0.00	83	1	714.63	714.63	0.00	43	1
E-n33-k4-s3-17	32	2	3	4	707.48*	707.48	707.48	0.00	80	0	709.26	707.48	0.00	39	0
E-n33-k4-s4-5	32	2	3	4	778.74*	778.74	778.74	0.00	79	1	778.74	778.74	0.00	40	1
E-n33-k4-s7-25	32	2	3	4	756.85*	756.85	756.85	0.00	82	0	760.16	756.85	0.00	40	0
E-n33-k4-s14-22	32	2	3	4	779.05*	779.05	779.05	0.00	82	0	779.05	779.05	0.00	44	0
Set 2b															
E-n51-k5-s2-4-17-46	50	4	4	5	530.76*	530.76	530.76	0.00	145	1	530.76	530.76	0.00	73	2
E-n51-k5-s2-17	50	2	3	5	597.49*	597.49	597.49	0.00	140	2	600.51	597.49	0.00	68	14
E-n51-k5-s4-46	50	2	3	5	530.76*	530.76	530.76	0.00	211	1	530.76	530.76	0.00	72	0
E-n51-k5-s6-12	50	2	3	5	554.81*	554.81	554.81	0.00	199	2	558.53	554.81	0.00	70	25
E-n51-k5-s6-12-32-37	50	4	4	5	531.92*	531.92	531.92	0.00	141	1	531.92	531.92	0.00	74	15
E-n51-k5-s11-19	50	2	3	5	581.64*	581.64	581.64	0.00	131	6	581.64	581.64	0.00	57	5
E-n51-k5-s11-19-27-47	50	4	4	5	527.63*	527.63	527.63	0.00	137	1	530.25	527.63	0.00	66	20
E-n51-k5-s27-47	50	2	3	5	538.22*	538.22	538.22	0.00	237	2	539.15	538.22	0.00	71	25
E-n51-k5-s32-37	50	2	3	5	552.28*	552.28	552.28	0.00	263	1	558.60	552.28	0.00	72	3
Set 2c															
E-n51-k5-s2-4-17-46	50	2	3	5	601.39*	601.39	601.39	0.00	209	4	601.39	601.39	0.00	71	1
E-n51-k5-s2-17	50	2	3	5	601.39*	601.39	601.39	0.00	141	1	601.39	601.39	0.00	72	1
E-n51-k5-s4-46	50	2	3	5	702.33*	703.37	702.33	0.00	196	4	711.18	708.14	0.83	63	12
E-n51-k5-s6-12	50	2	3	5	567.42*	567.42	567.42	0.00	129	3	574.00	572.27	0.86	65	21
E-n51-k5-s6-12-32-37	50	4	4	5	567.42*	567.42	567.42	0.00	136	2	568.39	567.42	0.00	66	24
E-n51-k5-s11-19	50	2	3	5	617.42*	617.42	617.42	0.00	137	1	617.42	617.42	0.00	65	0
E-n51-k5-s11-19-27-47	50	4	4	5	530.76*	530.76	530.76	0.00	135	1	530.76	530.76	0.00	69	1
E-n51-k5-s27-47	50	2	3	5	530.76*	530.76	530.76	0.00	134	0	530.76	530.76	0.00	62	0
E-n51-k5-s32-37	50	2	3	5	752.59*	752.60	752.60	0.00	132	5	752.60	752.60	0.00	62	7
Set 3a															
E-n22-k4-s13-14	21	2	3	4	526.15*	526.15	526.15	0.00	70	0	526.15	526.15	0.00	25	0
E-n22-k4-s13-16	21	2	3	4	521.09*	521.09	521.09	0.00	64	0	521.09	521.09	0.00	27	0
E-n22-k4-s13-17	21	2	3	4	496.38*	496.38	496.38	0.00	65	0	496.38	496.38	0.00	25	0
E-n22-k4-s14-19	21	2	3	4	498.80*	498.80	498.80	0.00	63	0	498.80	498.80	0.00	25	0
E-n22-k4-s17-19	21	2	3	4	512.80*	512.81	512.81	0.00	72	0	512.81	512.81	0.00	25	1
E-n22-k4-s19-21	21	2	3	4	520.42*	520.42	520.42	0.00	62	0	520.42	520.42	0.00	24	1
E-n33-k4-s16-22	32	2	3	4	672.17*	672.17	672.17	0.00	114	2	672.17	672.17	0.00	38	9
E-n33-k4-s16-24	32	2	3	4	666.02*	666.02	666.02	0.00	106	1	666.02	666.02	0.00	39	1
E-n33-k4-s19-26	32	2	3	4	680.36*	680.37	680.37	0.00	79	0	680.37	680.37	0.00	41	0
E-n33-k4-s22-26	32	2	3	4	680.37*	680.37	680.37	0.00	136	1	680.37	680.37	0.00	40	1
E-n33-k4-s24-28	32	2	3	4	670.43*	670.43	670.43	0.00	77	1	670.43	670.43	0.00	39	1
E-n33-k4-s25-28	32	2	3	4	650.58*	650.58	650.58	0.00	112	1	650.58	650.58	0.00	41	13

Table 7
Results part II/II for the 2E-VRP benchmark instances of Breunig et al. (2016). **: We found during the development of the heuristic a new BKS to instance 200-10-1 with objective 1553.75.

Instance	$\lvert\mathcal{V}^{\mathcal{C}}\rvert$	$\lvert\mathcal{V}^{\mathcal{S}}\rvert$	m^1	m^2	BKS	ALNS 4-threads					ALNS single-threaded				
						Avg. 5	Best	Gap BKS (%)	$t(s)$	$t^*(s)$	Avg. 5	Best	Gap BKS (%)	$t(s)$	$t^*(s)$
Set 3b & 3c															
E-n51-k5-s12-18	50	2	3	5	690.59*	690.59	690.59	0.00	125	10	697.61	692.56	0.29	69	12
E-n51-k5-s12-41	50	2	3	5	683.05*	683.05	683.05	0.00	239	29	694.69	683.05	0.00	64	8
E-n51-k5-s12-43	50	2	3	5	710.41*	710.41	710.41	0.00	137	1	710.41	710.41	0.00	62	0
E-n51-k5-s13-19	50	2	3	5	560.73*	560.73	560.73	0.00	119	2	562.96	560.73	0.00	74	0
E-n51-k5-s13-42	50	2	3	5	564.45*	564.45	564.45	0.00	174	1	564.45	564.45	0.00	64	0
E-n51-k5-s13-44	50	2	3	5	564.45*	564.45	564.45	0.00	121	1	564.45	564.45	0.00	65	0
E-n51-k5-s39-41	50	2	3	5	728.54*	728.54	728.54	0.00	179	9	728.54	728.54	0.00	65	19
E-n51-k5-s40-41	50	2	3	5	723.75*	723.75	723.75	0.00	129	30	730.15	728.54	0.66	57	21
E-n51-k5-s40-42	50	2	3	5	746.31*	746.31	746.31	0.00	122	13	752.47	746.31	0.00	65	19
E-n51-k5-s40-43	50	2	3	5	752.15*	752.15	752.15	0.00	242	9	755.35	752.15	0.00	61	16
E-n51-k5-s41-42	50	2	3	5	771.56*	771.56	771.56	0.00	203	11	771.56	771.56	0.00	63	15
E-n51-k5-s41-44	50	2	3	5	802.91*	804.83	802.91	0.00	169	32	817.86	810.52	0.95	67	18
Set 5															
100-5-1	100	5	5	32	1564.46*	1623.58	1605.59	2.63	554	358	1609.02	1599.61	2.25	166	119
100-5-1b	100	5	5	15	1108.62	1131.74	1121.70	1.18	699	303	1138.61	1125.23	1.50	133	19
100-5-2	100	5	5	32	1016.32*	1025.07	1022.14	0.57	647	261	1027.74	1025.50	0.90	200	123
100-5-2b	100	5	5	15	782.25	786.78	782.25	0.00	345	230	787.83	782.25	0.00	122	85
100-5-3	100	5	5	30	1045.29*	1047.88	1046.05	0.07	565	239	1051.54	1047.99	0.26	165	88
100-5-3b	100	5	5	16	828.54	830.16	828.99	0.05	372	175	830.89	828.99	0.05	128	58
100-10-1	100	10	5	35	1124.93	1126.07	1125.00	0.01	643	363	1139.15	1130.23	0.47	207	130
100-10-1b	100	10	5	18	916.25	923.85	921.29	0.55	402	119	925.46	924.58	0.91	152	27
100-10-2	100	10	5	35	990.58	1015.37	1010.72	2.03	454	197	1025.34	1022.21	3.19	237	175
100-10-2b	100	10	5	18	768.61	777.90	775.24	0.86	384	252	781.85	775.24	0.86	181	136
100-10-3	100	10	5	32	1043.25	1061.35	1053.02	0.94	810	401	1066.32	1053.99	1.03	142	105
100-10-3b	100	10	5	17	850.92	864.78	859.24	0.98	682	282	869.27	861.00	1.18	146	76
200-10-1**	100	10	5	62	1556.79	1610.73	1570.88	0.91	2375	1350	1614.53	1575.69	1.21	432	312
200-10-1b	100	10	5	30	1187.62	1212.92	1195.12	0.63	827	456	1230.18	1208.91	1.79	298	258
200-10-2	100	10	5	63	1365.74	1422.64	1389.11	1.71	1364	924	1443.59	1438.29	5.31	451	325
200-10-2b	100	10	5	30	1002.85	1018.89	**1002.63**	**-0.02**	810	482	1034.13	1020.14	1.72	259	200
200-10-3	100	10	5	63	1787.73	1887.67	1837.62	2.79	1096	776	1907.15	1895.00	6.00	450	246
200-10-3b	100	10	5	30	1197.90	1226.07	1219.92	1.84	825	340	1236.28	1222.47	2.05	380	313
Avg. Set 2					**578.27**	**578.30**	**578.27**	**0.00**	**135**	**1**	**579.54**	**578.62**	**0.06**	**54**	**6**
Avg. Set 3					**641.44**	**641.52**	**641.44**	**0.00**	**124**	**6**	**643.59**	**642.03**	**0.08**	**49**	**6**
Avg. Set 5					**1118.81**	**1144.08**	**1131.47**	**0.98**	**770**	**417**	**1151.05**	**1140.96**	**1.71**	**236**	**155**
Avg.					**734.46**	**740.82**	**737.63**	**0.25**	**290**	**107**	**743.77**	**740.35**	**0.48**	**98**	**43**

Table 8
Results part I/II for the SFL-VRP benchmark instances of Veenstra et al. (2018). Best Known Solutions (BKS) that are marked with an * are proven to be optimal.

Inst.	$\lvert\mathcal{V}^{\mathcal{C}}\rvert$	$\lvert\mathcal{V}^{\mathcal{L}}\rvert$	Veenstra et al. (2018)			BKS	ALNS heuristic multi-threaded						ALNS heuristic single-threaded					
			Avg. 5	Best	$t(s)$		Avg. 5	Best	Gap Veenstra (%)	Gap BKS (%)	$t(s)$	$t^*(s)$	Avg. 5	Best	Gap Veenstra (%)	Gap BKS (%)	$t(s)$	$t^*(s)$
R001	30	10	3884.0	3884	10.4	3884*	3884.0	3884	0.00	0.00	71	0	3884.0	3884	0.00	0.00	36	0
R002	30	15	2897.0	2897	11.1	2897*	2897.0	2897	0.00	0.00	92	0	2897.0	2897	0.00	0.00	32	0
R003	30	20	2050.0	2050	11.3	2050*	2050.0	2050	0.00	0.00	77	0	2050.0	2050	0.00	0.00	38	0
R004	30	25	3496.0	3496	12.0	3496*	3496.0	3496	0.00	0.00	135	5	3506.0	3496	0.00	0.00	39	7
R005	30	30	3841.0	3841	15.0	3841*	3841.0	3841	0.00	0.00	84	0	3841.0	3841	0.00	0.00	43	1
R006	30	35	2767.0	2767	17.0	2767*	2767.0	2767	0.00	0.00	92	0	2767.0	2767	0.00	0.00	42	0
R007	30	40	2129.0	2129	15.9	2129*	2130.6	2129	0.00	0.00	101	7	2133.6	2129	0.00	0.00	46	16
R008	30	45	1012.4	1012	17.2	1012*	1012.0	1012	0.00	0.00	101	8	1012.4	1012	0.00	0.00	52	10
R009	30	50	455.0	455	18.2	455*	456.0	455	0.00	0.00	108	3	455.2	455	0.00	0.00	56	31
R010	40	10	4388.0	4388	19.7	4388*	4388.0	4388	0.00	0.00	91	0	4388.0	4388	0.00	0.00	50	0
R011	40	15	3874.0	3874	19.4	3874*	3874.0	3874	0.00	0.00	83	0	3874.0	3874	0.00	0.00	40	0
R012	40	20	3531.0	3531	16.7	3531*	3531.0	3531	0.00	0.00	113	0	3531.0	3531	0.00	0.00	37	0
R013	40	25	4495.0	4495	21.5	4495*	4495.0	4495	0.00	0.00	97	0	4495.0	4495	0.00	0.00	46	1
R014	40	30	4399.0	4399	25.7	4399*	4399.0	4399	0.00	0.00	143	0	4399.0	4399	0.00	0.00	54	0
R015	40	35	3964.0	3964	27.0	3964*	3964.0	3964	0.00	0.00	106	3	3964.0	3964	0.00	0.00	53	10
R016	40	40	2789.0	2789	27.5	2789*	2795.8	2789	0.00	0.00	118	11	2802.6	2789	0.00	0.00	56	27
R017	40	45	2641.0	2641	24.9	2641*	2641.0	2641	0.00	0.00	105	1	2641.0	2641	0.00	0.00	50	1
R018	40	50	3390.0	3390	29.8	3390*	3390.0	3390	0.00	0.00	115	3	3390.0	3390	0.00	0.00	59	9
R019	50	10	5259.0	5259	31.2	5259*	5259.0	5259	0.00	0.00	158	0	5259.0	5259	0.00	0.00	55	0

(continued on next page)

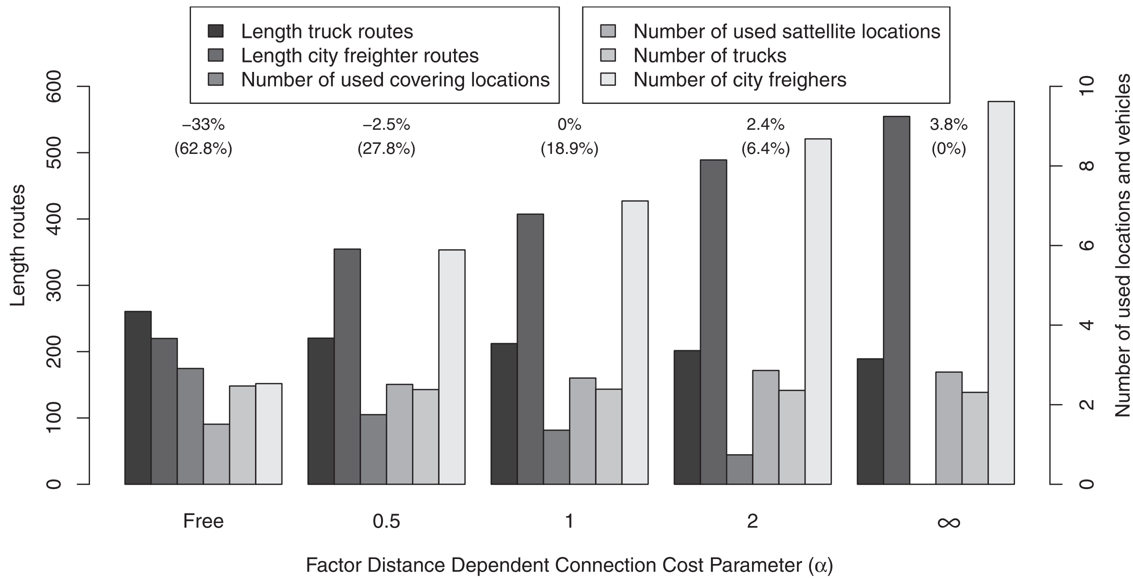
Table 8 (*continued*)

Inst.	$\lvert\mathcal{V}^{\mathcal{C}}\rvert$	$\lvert\mathcal{V}^{\mathcal{L}}\rvert$	Veenstra et al. (2018)			BKS	ALNS heuristic multi-threaded						ALNS heuristic single-threaded					
			Avg. 5	Best	$t(s)$		Avg. 5	Best	Gap Veenstra (%)	Gap BKS (%)	$t(s)$	$t^*(s)$	Avg. 5	Best	Gap Veenstra (%)	Gap BKS (%)	$t(s)$	$t^*(s)$
R020	50	15	4768.0	4768	34.0	4768*	4768.0	4768	0.00	0.00	94	1	4768.0	4768	0.00	0.00	49	1
R021	50	20	5058.0	5058	32.3	5058*	5058.0	5058	0.00	0.00	182	1	5058.0	5058	0.00	0.00	60	1
R022	50	25	2351.0	2351	30.5	2351*	2351.0	2351	0.00	0.00	99	1	2351.0	2351	0.00	0.00	46	3
R023	50	30	4040.0	4040	38.2	4040*	4040.0	4040	0.00	0.00	180	2	4040.0	4040	0.00	0.00	62	2
R024	50	35	3959.0	3959	38.3	3959*	3959.0	3959	0.00	0.00	173	2	3959.0	3959	0.00	0.00	67	4
R025	50	40	3697.0	3697	35.7	3697*	3703.4	3697	0.00	0.00	210	86	3699.0	3697	0.00	0.00	63	29
R026	50	45	4120.0	4120	42.7	4120*	4120.0	4120	0.00	0.00	120	1	4120.0	4120	0.00	0.00	66	5
R027	50	50	2715.2	2715	38.1	2715*	2715.0	2715	0.00	0.00	149	6	2715.0	2715	0.00	0.00	59	8
R028	60	10	6314.5	6245	58.9	6170*	6170.0	6170	-1.20	0.00	217	0	6170.0	6170	-1.20	0.00	111	0
R029	60	15	4289.0	4289	42.9	4289*	4289.0	4289	0.00	0.00	108	0	4289.0	4289	0.00	0.00	62	0
R030	60	20	5245.0	5245	54.1	5245*	5245.0	5245	0.00	0.00	118	2	5246.0	5245	0.00	0.00	66	6
R031	60	25	2540.0	2540	41.0	2540*	2540.0	2540	0.00	0.00	117	1	2540.0	2540	0.00	0.00	55	4
R032	60	30	3854.0	3854	49.0	3854*	3854.0	3854	0.00	0.00	114	2	3854.0	3854	0.00	0.00	64	2
R033	60	35	2960.0	2960	45.4	2960*	2960.0	2960	0.00	0.00	176	29	2960.0	2960	0.00	0.00	59	4
R034	60	40	4830.0	4830	71.5	4830*	4830.0	4830	0.00	0.00	214	2	4830.0	4830	0.00	0.00	81	2
R035	60	45	3562.0	3562	55.4	3562	3562.0	3562	0.00	0.00	137	3	3562.0	3562	0.00	0.00	77	6
R036	60	50	2677.0	2677	48.1	2677*	2677.0	2677	0.00	0.00	127	20	2677.0	2677	0.00	0.00	65	12

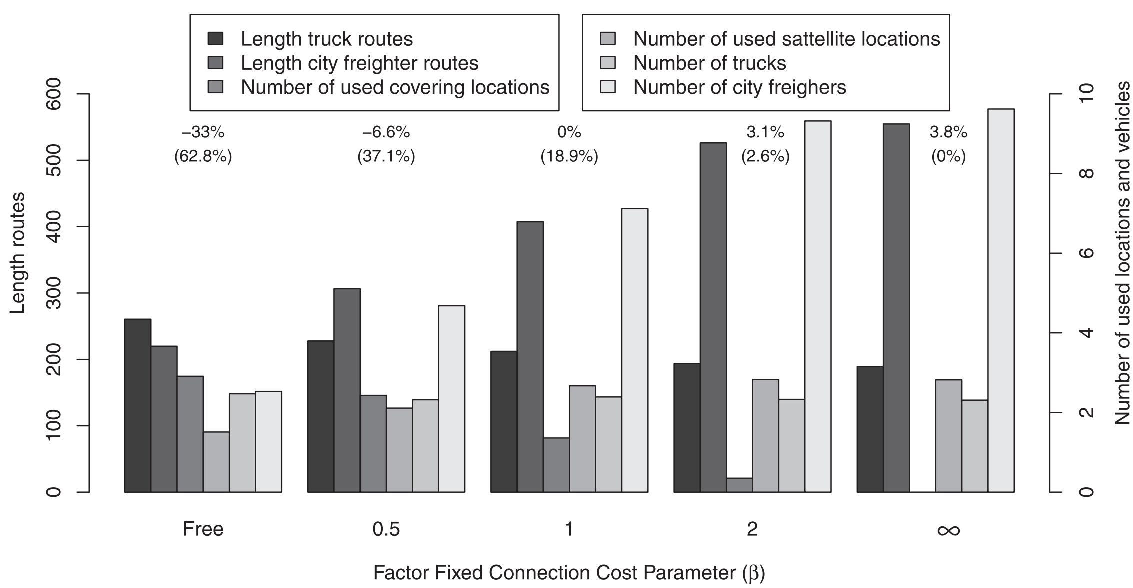
Table 9
Results part II/II for the SFL-VRP benchmark instances of Veenstra et al. (2018). Best Known Solutions (BKS) that are marked with an * are proven to be optimal.

Inst.	$\lvert\mathcal{V}^{\mathcal{C}}\rvert$	$\lvert\mathcal{V}^{\mathcal{L}}\rvert$	Veenstra et al. (2018)			BKS	ALNS heuristic multi-threaded						ALNS heuristic single-threaded					
			Avg. 5	Best	$t(s)$		Avg. 5	Best	Gap Veenstra (%)	Gap BKS (%)	$t(s)$	$t^*(s)$	Avg. 5	Best	Gap Veenstra (%)	Gap BKS (%)	$t(s)$	$t^*(s)$
R037	70	10	5605.0	5605	81.8	5605*	5605.0	5605	0.00	0.00	146	0	5605.0	5605	0.00	0.00	90	0
R038	70	15	5157.0	5157	73.4	5157*	5157.0	5157	0.00	0.00	156	0	5157.0	5157	0.00	0.00	60	0
R039	70	20	5447.0	5447	76.1	5447*	5447.0	5447	0.00	0.00	120	1	5447.0	5447	0.00	0.00	71	1
R040	70	25	4738.0	4738	118.2	4738*	4738.0	4738	0.00	0.00	120	1	4738.0	4738	0.00	0.00	63	2
R041	70	30	5083.0	5083	92.6	5083*	5083.0	5083	0.00	0.00	140	1	5083.0	5083	0.00	0.00	71	6
R042	70	35	1732.0	1732	104.0	1732*	1732.0	1732	0.00	0.00	116	1	1732.0	1732	0.00	0.00	55	2
R043	70	40	4053.0	4053	96.8	4053	4053.4	4053	0.00	0.00	132	6	4053.4	4053	0.00	0.00	73	38
R044	70	45	3365.0	3365	90.7	3365	3376.4	3365	0.00	0.00	130	21	3376.4	3365	0.00	0.00	73	40
R045	70	50	2287.0	2286	13.3	2286	2293.8	2286	0.00	0.00	140	11	2297.2	2286	0.00	0.00	70	28
R046	80	10	6244.0	6244	108.4	6244*	6244.0	6244	0.00	0.00	194	0	6244.0	6244	0.00	0.00	87	0
R047	80	15	5162.0	5162	99.3	5162	5162.0	5162	0.00	0.00	119	1	5162.0	5162	0.00	0.00	62	1
R048	80	20	5997.0	5997	118.6	5997	5997.0	5997	0.00	0.00	192	2	5997.0	5997	0.00	0.00	95	3
R049	80	25	4323.0	4323	92.6	4323	4324.8	4323	0.00	0.00	125	5	4324.8	4323	0.00	0.00	74	21
R050	80	30	4155.0	4155	104.2	4145*	4145.0	4145	−0.24	0.00	122	1	4145.0	4145	−0.24	0.00	71	3
R051	80	35	6181.0	6181	96.8	6181	6181.0	6181	0.00	0.00	251	12	6181.0	6181	0.00	0.00	105	19
R052	80	40	4065.0	4065	90.7	4065*	4065.0	4065	0.00	0.00	160	5	4065.0	4065	0.00	0.00	92	17
R053	80	45	2424.0	2424	123.3	2424*	2424.0	2424	0.00	0.00	141	12	2426.0	2424	0.00	0.00	78	29
R054	80	50	3623.0	3623	108.4	3623	3623.0	3623	0.00	0.00	303	70	3630.6	3623	0.00	0.00	98	17
R055	90	10	7453.0	7453	172.5	7453	7453.0	7453	0.00	0.00	201	7	7453.0	7453	0.00	0.00	146	29
R056	90	15	5919.0	5919	139.0	5919	5919.0	5919	0.00	0.00	145	1	5919.0	5919	0.00	0.00	94	1
R057	90	20	4668.0	4668	111.2	4668	4668.0	4668	0.00	0.00	111	0	4668.0	4668	0.00	0.00	59	1
R058	90	25	6003.0	6003	136.7	6003	6003.0	6003	0.00	0.00	199	1	6003.0	6003	0.00	0.00	94	13
R059	90	30	4638.0	4638	130.5	4638	4638.6	4638	0.00	0.00	132	29	4639.8	4638	0.00	0.00	73	11
R060	90	35	4867.0	4867	160.0	4867	4870.0	4867	0.00	0.00	223	13	4867.0	4867	0.00	0.00	91	6
R061	90	40	3681.0	3681	110.2	3681	3681.0	3681	0.00	0.00	134	18	3683.0	3681	0.00	0.00	81	16
R062	90	45	2553.0	2553	133.9	2553	2544.6	**2543**	−0.39	−0.39	144	23	2549.4	2543	−0.39	−0.39	84	9
R063	90	50	1586.0	1586	124.8	1586*	1588.8	1586	0.00	0.00	149	12	1594.6	1586	0.00	0.00	87	49
R064	100	10	7034.0	7034	218.8	7034	7034.0	7034	0.00	0.00	232	3	7034.0	7034	0.00	0.00	163	19
R065	100	15	7389.0	7389	216.9	7389	7389.0	7389	0.00	0.00	212	37	7439.0	7439	0.68	0.68	108	3
R066	100	20	5790.0	5790	186.1	5790	5802.6	5790	0.00	0.00	151	25	5811.0	5811	0.36	0.36	99	4
R067	100	25	5720.0	5720	178.3	5720	5720.0	5720	0.00	0.00	159	1	5720.0	5720	0.00	0.00	103	4
R068	100	30	3798.0	3798	139.1	3798	3798.0	3798	0.00	0.00	187	1	3798.0	3798	0.00	0.00	64	2
R069	100	35	3481.6	3477	163.6	3477	3477.0	3477	0.00	0.00	147	7	3477.0	3477	0.00	0.00	81	18
R070	100	40	3621.0	3621	149.1	3621	3629.8	3621	0.00	0.00	150	27	3627.6	3621	0.00	0.00	83	12
R071	100	45	4621.0	4621	211.4	4621	4621.0	4621	0.00	0.00	326	7	4621.0	4621	0.00	0.00	102	7
R072	100	50	873.0	873	146.9	873*	873.0	873	0.00	0.00	218	20	873.0	873	0.00	0.00	76	24
Avg.			**4049.7**	**4048.6**	**78.4**	**4047.5**	**4048.2**	**4047.3**	**-0.03**	**-0.01**	**147**	**8**	**4049.6**	**4048.3**	**-0.01**	**0.01**	**70**	**9**

Appendix B. Additional figures belonging to Section 5.

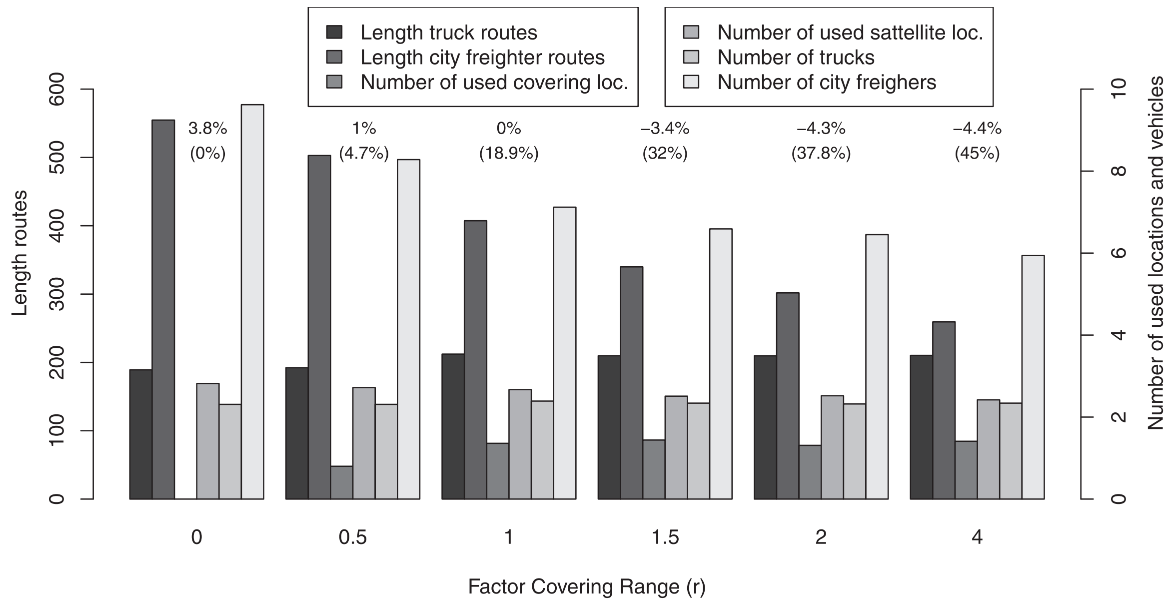


(a) The effect of changing the distance dependent connection cost parameter α.

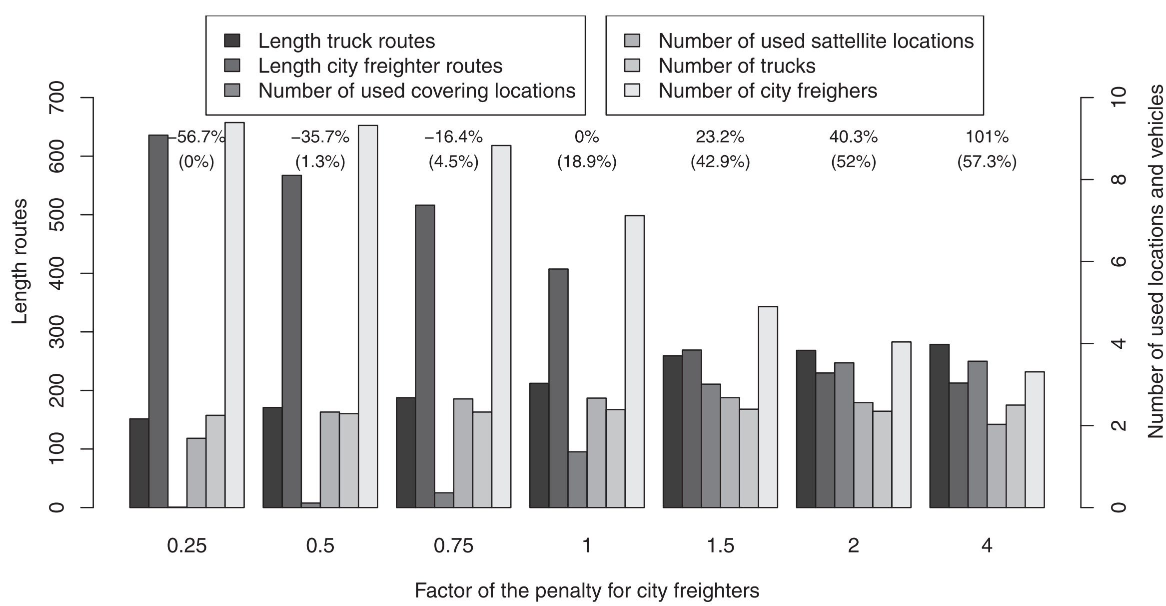


(b) The effect of changing the fixed connection cost parameter β

Fig. 4. The average solution characteristics for different parameter settings.



(a) The effect of changing the covering range parameter r



(b) The effect of changing cargo bike driving costs.

Fig. 5. The average solution characteristics for different parameter settings.

References

Anderluh, A., Hemmelmayr, V.C., Nolz, P.C., 2017. Synchronizing vans and cargo bikes in a city distribution network. Cent. Eur. J. Oper. Res. 25 (2), 345–376.

Arnold, F., Cardenas, I., Sörensen, K., Dewulf, W., 2018. Simulation of b2c e-commerce distribution in antwerp using cargo bikes and delivery points. Eur. Transp. Res. Rev. 10 (1), 2.

Breunig, U., Baldacci, R., Hartl, R.F., Vidal, T., 2019. The electric two-echelon vehicle routing problem. Comput. Oper. Res. 103, 198–210.

Breunig, U., Schmid, V., Hartl, R.F., Vidal, T., 2016. A large neighbourhood based heuristic for two-echelon routing problems. Comput. Oper. Res. 76, 208–225.

Cattaruzza, D., Absi, N., Feillet, D., González-Feliu, J., 2017. Vehicle routing problems for city logistics. EURO J. Transp. Logist. 6 (1), 51–79.

Cuda, R., Guastaroba, G., Speranza, M.G., 2015. A survey on two-echelon routing problems. Comput. Oper. Res. 55, 185–199.

Demir, E., Huang, Y., Scholts, S., Van Woensel, T., 2015. A selected review on the negative externalities of the freight transportation: modeling and pricing. Transp. Res. Part E 77, 95–114.

Deutsch, Y., Golany, B., 2018. A parcel locker network as a solution to the logistics last mile problem. Int. J. Prod. Res. 56 (1–2), 251–261.

Franceschetti, A., Demir, E., Honhon, D., Van Woensel, T., Laporte, G., Stobbe, M., 2017. A metaheuristic for the time-dependent pollution-routing problem. Eur. J. Oper. Res. 259 (3), 972–991.

Grangier, P., Gendreau, M., Lehuédé, F., Rousseau, L.-M., 2016. An adaptive large neighborhood search for the two-echelon multiple-trip vehicle routing problem with satellite synchronization. Eur. J. Oper. Res. 254 (1), 80–91.

Guastaroba, G., Speranza, M.G., Vigo, D., 2016. Intermediate facilities in freight transportation planning: a survey. Transp. Sci. 50 (3), 763–789.

Hemmelmayr, V.C., Cordeau, J.-F., Crainic, T.G., 2012. An adaptive large neighborhood search heuristic for two-echelon vehicle routing problems arising in city logistics. Comput. Oper. Res. 39 (12), 3215–3228.

Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P., 1983. Optimization by simulated annealing. Science 220 (4598), 671–680.

Perboli, G., Tadei, R., 2010. New families of valid inequalities for the two-echelon vehicle routing problem. Electron. Notes Discret. Math. 36, 639–646.

Perboli, G., Tadei, R., Vigo, D., 2011. The two-echelon capacitated vehicle routing problem: models and math-based heuristics. Transp. Sci. 45 (3), 364–380.

Pisinger, D., Ropke, S., 2007. A general heuristic for vehicle routing problems. Comput. Oper. Res. 34 (8), 2403–2435.

Prodhon, C., Prins, C., 2014. A survey of recent research on location-routing problems. Eur. J. Oper. Res. 238 (1), 1–17.

Ropke, S., Pisinger, D., 2006. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. Transp. Sci. 40 (4), 455–472.

Savelsbergh, M., Van Woensel, T., 2016. 50th anniversary invited articlecity logistics: challenges and opportunities. Transp. Sci. 50 (2), 579–590.

Schrotenboer, A.H., Uit het Broek, M.A.J., Jargalsaikhan, B., Roodbergen, K.J., 2018. Coordinating technician allocation and maintenance routing for offshore wind farms. Comput. Oper. Res. 98, 185–197.

Schrotenboer, A. H., Wruck, S., Vis, I. F., Roodbergen, K. J., 2019. Integration of returns and decomposition of customer orders in e-commerce warehouses. arXiv preprint arXiv:1909.01794.

Stenger, A., Schneider, M., Schwind, M., Vigo, D., 2012. Location routing for small package shippers with subcontracting options. Int. J. Prod. Econ. 140 (2), 702–712.

Veenstra, M., Roodbergen, K.J., Coelho, L.C., Zhu, S.X., 2018. A simultaneous facility location and vehicle routing problem arising in health care logistics in the netherlands. Eur. J. Oper. Res. 268 (2), 703–715.

Wang, K., Shao, Y., Zhou, W., 2017. Matheuristic for a two-echelon capacitated vehicle routing problem with environmental considerations in city logistics service. Transp. Res. Part D 57, 262–276.

Zhou, L., Baldacci, R., Vigo, D., Wang, X., 2018. A multi-depot two-echelon vehicle routing problem with delivery options arising in the last mile distribution. Eur. J. Oper. Res.. 265 (2), 765–778.

Zhou, L., Wang, X., Ni, L., Lin, Y., 2016. Location-routing problem with simultaneous home delivery and customers pickup for city distribution of online shopping purchases. Sustainability 8 (8), 828.