DISSERTATION / DOCTORAL THESIS

Titel der Dissertation / Title of the Doctoral Thesis

„Metaheuristics for two-echelon routing problems“

verfasst von / submitted by

Mag. Ulrich Breunig

angestrebter akademischer Grad / in partial fulfilment of the requirements for the degree of

Doctor of Philosophy (PhD)

Wien, 2022 / Vienna 2022

Studienkennzahl lt. Studienblatt / degree programme code as it appears on the student record sheet:	UA 794 370 403
Studienrichtung lt. Studienblatt / field of study as it appears on the student record sheet:	Logistics and Operations Management
Betreut von / Supervisor:	O.Univ.Prof. Dipl-Ing. Dr. Richard F. Hartl

Preface

This dissertation has been prepared over several years and represents the result of my own work, except the Section 3.4, which was programmed and written mainly by the corresponding co-author of the paper, Roberto Baldacci.

Co-author Verena Schmid helped with the problem formulation and the basic programming framework for the first paper, Thibaut Vidal always brought good input on other points of view and techniques to explore the problem further. Together with my supervising professor Richart F. Hartl we've had many iterations of discussions on analysing results, interpreting them and subsequently reinforcing the solution algorithms.

The programming code for the heuristic solution algorithms, written in C++ and Java, was implemented by me.

Many contents of this book are already published in whole or in parts in following scientific journals: Breunig et al. (2016, 2017, 2019).

Abstract

Two-echelon distribution systems are attractive from an economical standpoint and help to keep large vehicles out of densely populated city centers. Large trucks can be used to deliver goods to intermediate facilities in accessible locations, whereas smaller vehicles allow to reach the final customers. This two-tiered setup comes of course at a cost of higher planning complexity. This thesis is composed of the publications Breunig et al. (2016, 2017, 2019) and studies large neighbourhood based metaheuristics for the classic two-echelon vehicle routing problem, the classic two-echelon location routing problem, and introduces a new problem variant, which is called the electric two-echelon vehicle routing problem.

For the first two variants we can show that our developed metaheuristic outperforms the ones from previous literature. The latter takes into account specific characteristics of battery-powered vehicles to be used for the second echelon, reducing noise and pollution in city centers. We designed representative sets of new benchmark instances simulating realistic metropolitan areas. In particular, we observe that the detour miles due to recharging decrease proportionally to $1/\rho^x$ with $x \approx 5/4$ as a function of the charging stations density ρ; e.g., in a scenario where the density of charging stations is doubled, recharging detours are reduced by 58%. Finally, we evaluate the trade-off between battery capacity and detour miles. This estimate is critical for strategic fleet-acquisition decisions, in a context where large batteries are generally more costly and less environment-friendly.

German Abstract

Zweistufige Transportsysteme können ökonomisch vorteilhaft sein, auch kann man damit in Stadtzentren mit hoher Bevölkerungsdichte große Fahrzeuge vermeiden. Die großen Fahrzeuge transportieren die Güter nur bis zu Zwischenlagern am Stadtrand, ab dort übernehmen kleinere Fahrzeuge die letzten Kilometer bis zum Endkunden. Solche zweistufigen Systeme bringen natürlich Kosten mit sich: Die kostensparende Planung der Routen ist komplexer und aufwändiger. Diese Dissertation setzt sich aus den Publikationen von Breunig et al. (2016, 2017, 2019) zusammen. Die Metaheuristiken wurden für das klassische two-echelon vehicle routing problem und das klassische two-echelon location routing problem entwickelt und zeigen bessere Leistungen als zuvor veröffentlichte Lösungsmethoden. Weiters wurde ein neues Problem eingeführt: das electric two-echelon vehicle routing problem. Bei diesem werden im Stadtinneren elektrische Fahrzeuge verwendet, deren spezielle Eigenschaften berücksichtigt werden müssen. Wir haben anhand neu entwickelter, repräsentativer Testinstanzen untersucht, welche Auswirkungen die Reichweite der Batterie auf die Kosten für Umwege zu Ladestationen hat. Diese Schätzungen helfen bei strategischen Entscheidungen bei der Auswahl von Fahrzeugflotten.

Contents

List of Figures

List of Tables

List of Algorithms

1 Introduction

Transportation of goods has an increasing significance in our society. Digitalisation, cheap transport and people's convenience lead to more and more goods being shipped to end-customers. But also stores need their products physically available - for customers, who don't want to shop online.

On the other hand trucks are still a nuisance, especially within city limits. They emit noise and pollution, and may cause congestions; parking space is scarce.

This dissertation deals with ways to plan truck routes efficiently; mostly in terms of City Logistic schemes, where different types of vehicles are used. The basic idea is to use large trucks for longer distances outside of cities, and re-load the goods to a different vehicle type in small facilities outside the cities. These are preferably smaller and more suitable for inner city deliveries, and in some cases even battery electric vehicles - with their idiosyncratic restrictions.

The work is structured as follows: first, we present the first part and publication, dealing with the classic two-echelon vehicle routing problem (2EVRP) and two-echelon location routing problem (2ELRP). The second part was published in another paper and extends these classic problems by introducing battery electric vehicles on the second echelon. Afterwards, we will give an overview of the more recent publications from the literature; an update to what has happened in the scientific literature since the before mentioned papers were published.

2 A large neighbourhood-based heuristic for two-echelon routing problems

This part was published as Breunig et al. (2016).

2.1 Introduction

The traffic of vehicles is a major nuisance in densely populated areas. Trucks disturb peoples' well-being by emitting noise and air pollution. As the amount of goods in transit increases, a proper planning of road networks and facility locations becomes critical to mitigate congestion. To face these challenges, algorithmic tools have been developed to optimise city logistics at several levels: considering traffic regulation, itineraries and network design choices. Boosting the efficiency of goods transportation from suppliers to customers presents important challenges for different planning horizons. On the operational level, efficient itineraries must be found for the available vehicles from day to day, e.g., reducing the travelled distance. On a tactical level, the overall fleet size, vehicle dimensions, capacities and characteristics are of interest. Larger trucks are more efficient in terms of cost per shipped quantity, whereas smaller vehicles are more desirable in city centres: they emit less noise, and only need smaller parking spots. Finally, the clever selection of locations for production sites, warehouses, and freight terminals is a typical strategic decision.

In this article, we consider the problem of jointly determining good routes to deliver goods to customers, and at which intermediate facilities a switch from larger trucks to smaller city freighters should happen. This problem is challenging, due to the combination of these two families of combinatorial decisions.

To address this problem, we propose a simple metaheuristic, which combines local and large neighbourhood search with the ruin and recreate principle. The method is conceptually simple and fast, exploiting a limited subset of neighbourhoods in combination with a simple strategy for closing and opening intermediate facilities. We conduct extensive computational experiments on 2EVRP and 2ELRPSD instances to investigate the contribution of these operators, and measure the performance of the method on both problem classes. For the 2EVRP, this algorithm is able to reach or outperform 95% of the best known solutions. In general, for both problems, high-quality solutions are attained in short computing times. As such, this algorithm will serve as a good basis for future developments on more complex and realistic two-tiered delivery problems.

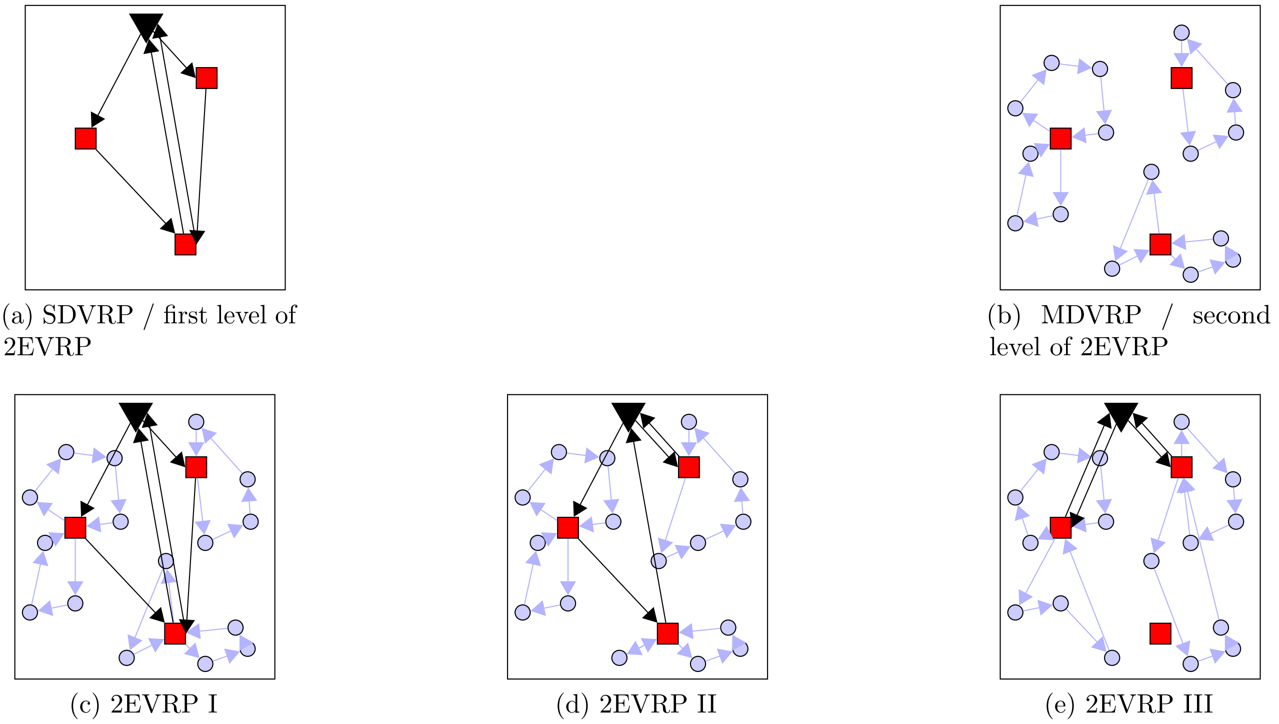
The paper is organised as follows. The problems are described in Section 2.2 and an overview of the related literature is given in Section 2.3. Mathematical formulations are presented in Section 2.4. Section 2.5 describes the proposed algorithm. Section 2.6 reviews current available benchmark instances and examines the performance of the proposed method. Section 2.7 concludes.

2.2 Problem description

Vehicle routing problems (VRPs) are a class of combinatorial optimisation problems, which aim to find good itineraries to service a number of customers with a fleet of vehicles. The 2EVRP is a variant of a VRP, which exploits the different advantages of small and large vehicles in an integrated delivery system. The goal is to design an efficient distribution chain, organised in two levels: trucks operate on the first level between a central depot and several selected intermediate distribution facilities, called satellites. The second level also includes the satellites – because both levels are interconnected there – as well as the end-customers. Small city freighters are operated between satellites and customers. The depot supplies sufficient quantities to satisfy all customer demands. The products are directly transferred from trucks to city freighters at satellite locations. These city freighters will perform the deliveries to the final customers. Any shipment or part of shipment has to transit through exactly one satellite, and the final delivery to the customer is done in one block. As such, split deliveries are not allowed for city freighters. The quantity ("demand") of goods shipped to a satellite is not explicitly given, but evaluated as the sum of all customer demands served with city freighters originating from this satellite. Depending on the second level itineraries, split deliveries can occur on the first level since the total quantity needed at one satellite can exceed the capacity of one truck.

Finding good combined decisions for routing and intermediate facility openings is significantly more difficult than in well-studied settings such as the capacitated vehicle routing problem (CVRP). The special case of a 2EVRP with only one satellite can be seen as a VRP (Cuda et al., 2015; Perboli et al., 2011). The first level of the 2EVRP reduces to a CVRP with split deliveries. The structure of the second level is a multi-depot vehicle routing problem (MDVRP), where the depots correspond to the satellite locations (Jepsen et al., 2012). Those two levels have to be synchronised with each other. The 2EVRP is a generalisation of the classical VRP and is thus NP-hard.

Figure 1 shows different set-ups for goods distribution. The depot is represented by a triangle, satellites by squares, and customers by circles. Figures 1a and 1b show graphical representations of a split delivery vehicle routing problem (SDVRP) and a MDVRP respectively. Figures 1c–1e represent feasible solutions for the 2EVRP: with split deliveries occurring at one of the satellites, without split deliveries, and in Figure 1e a solution where only a subset of satellites is used.



(a) SDVRP / first level of 2EVRP

(b) MDVRP / second level of 2EVRP

(c) 2EVRP I

(d) 2EVRP II

(e) 2EVRP III

Figure 1: Subproblems related to the 2EVRP and different solutions depending on which intermediate facilities are used

The proposed algorithm has primarily been designed for the 2EVRP, and then tested on the 2ELRPSD, which includes additional tactical decisions. The basic structure of the 2ELRPSD is very similar to the 2EVRP. The main difference is that it corresponds to a more tactical planning since only potential locations for depots or satellites are known and the use of any location results in opening costs. In contrast with the 2EVRP, the fleet size is unbounded, but fixed costs are counted for the use of each vehicle. The classical benchmark sets from the literature include different costs per mile for large first level trucks and small city freighters, unlike in 2EVRP benchmark instances, where mileage costs are identical for all vehicles. Finally, split deliveries are not allowed at both levels. We thus applied our algorithm to 2ELRPSD instances, where location decisions have to be taken at the secondary facilities. Following the notations of Boccia et al. (2011) we focus on $3/T/\overline{T}$ problems.

2.3 Literature Review

Jacobsen and Madsen (1980) were amongst the first to introduce a two-echelon distribution optimisation problem. They proposed a three stage heuristic to solve the daily distribution of newspapers in Denmark, but no mathematical model was designed. Several possible transfer points were considered to transfer newspapers from one vehicle to another. The solution to this problem consists of three layers of decisions: the number and location of transfer points, the tours from the printing office and the tours from the transfer points to the retailers. An improved solution algorithm for the same problem can be found in Madsen (1983). Following the nomenclature of the authors and the classification in the recent survey on two-echelon routing problems by Cuda et al. (2015), the problem includes location decisions. Nevertheless, from today's point of view, it cannot be categorised as a 2ELRP as there are no opening costs associated with the use of intermediate facilities and retailers can also be served directly from the printing office without using intermediate nodes. In addition, the problem cannot be categorised as a 2EVRP since first-level split deliveries are not allowed.

Two-echelon vehicle routing problem Crainic et al. (2004) used data from Rome to study an integrated urban freight management system. As large trucks cannot pass through the narrow streets in the city centre, they used intermediate facilities to redistribute loads from large trucks to smaller vehicles. The city was divided into several commercial and external zones, and a mathematical location-allocation formulation was proposed and solved using a commercial solver. A comparison between solutions for delivering goods lead to the conclusion that intermediate facilities reduce the use of large trucks significantly, and more work is done by smaller city freighters.

Crainic et al. (2009) formulated a time dependent version of the problem, including time windows at the customers. To our knowledge, there are no test instances or solution approaches for this variant so far. Crainic et al. (2010) studied the impact of different two-tiered transportation set-ups on total cost. According to their results, the 2EVRP can yield better solutions than the VRP if the depot is not located within the customer area but externally. Perboli et al. (2011) introduced a flow-based mathematical formulation and generated three sets of instances for the 2EVRP with a maximum of 50 customers and four satellites, based on VRP instances. Their branch-and-cut approach is able to solve instances with up to 21 customers to optimality. Perboli et al. (2010) solved additional instances and reduced the optimality gap on others by means of new cutting rules.

Crainic et al. (2011) solved the 2EVRP with a multi-start heuristic. The method first assigns customers to satellites heuristically, and then solves the remaining VRPs with an exact method. In a perturbation step, the assignment of customers to satellites is changed, then the problem is solved again, until a number of iterations is reached.

Jepsen et al. (2012) presented a branch-and-cut method, solving 47 out of 93 test instances to optimality, 34 of them for the first time. The authors have been the first to consider a constraint on the number of vehicles *per satellite*, although it was already specified before in the existing data set. This additional constraint had not been taken into account by previous publications.

Hemmelmayr et al. (2012) developed a metaheuristic based on adaptive large neighborhood search (ALNS) with a variety of twelve destroy and repair operators. This approach tends to privilege accuracy (high quality solutions) over simplicity and flexibility (Cordeau et al., 2002). The authors also introduced new larger test instances with up to 200 customers and five to ten satellites. Note that the results of ALNS on the problem instances with 50 customers cannot be compared with the proven optimal solutions by Jepsen et al. (2012), since the algorithm does not consider a limit on the number of vehicles per satellite, but rather a constraint on the total number of vehicles. For most problem instances, this algorithm found the current best known solution or improved it.

Santos et al. (2015) implemented a branch-and-cut-and-price algorithm, which relies on a reformulation of the problem to overcome symmetry issues. They also introduced several classes of valid inequalities. The algorithm performs well in comparison to other exact methods, and they reported solutions for instances with up to 50 customers.

Baldacci et al. (2013) presented a promising exact method to solve the 2EVRP. They decomposed the problem into a limited set of MDVRPs with side constraints. Detailed results and comparisons with previous publications were provided, as they considered both variants on the instances with 50 customers: with and without the constraint on vehicles per satellite. They also introduced a new set of instances with up to 100 customers.

Recently Zeng et al. (2014) published a greedy randomized adaptive search procedure, combined with a route-first cluster-second splitting algorithm and a variable neighbourhood descent. They also provide a mathematical formulation. They provide quite good results, although unfortunately their algorithm was tested only with instances comprising up to 50 customers.

Two-echelon location routing problem The capacitated 2ELRP is by far the most studied version of the 2ELRP. Many papers consider location decisions only at the second stage, either because the use of depots is an outcome of the first level routing optimisation, or they consider problems with only one single depot location which is known a priori (2ELRPSD).

Laporte (1988) presented a general analysis of location routing problems and multi-layered problem variants. They compared several mathematical formulations and their computational performance. In a slightly different context, Laporte and Nobert (1988) formulated a vehicle flow model for the 2ELRP. The locations of the depots are assumed to be fixed and unchangeable, such that the location decisions only occur for the satellites. Following the notation by Boccia et al. (2011) they analysed $3/R/\overline{R}$, $3/R/\overline{T}$, $3/T/\overline{R}$, and $3/T/\overline{T}$ problem settings.

Boccia et al. (2011) provided three mathematical formulations for the 2ELRP using one-, two-, and three indexed variables inspired from VRP and MDVRP formulations. A commercial solver was used to solve some instances generated by the authors, comparing two of the formulations in terms of speed and quality.

Nguyen et al. (2012a) introduced two new sets of instances for the 2ELRPSD. They implemented a GRASP with path relinking and a learning process and provided detailed results. In Nguyen et al. (2012b) the authors improved their findings on the same instances by using a multi-start iterated local search.

Contardo et al. (2012) proposed a branch-and-cut algorithm, which is based on a two-indexed vehicle flow formulation, as well as an ALNS heuristic. Both solution approaches were applied to one set of 2ELRP instances, and two sets of 2ELRPSD instances, outperforming previous heuristics.

Schwengerer et al. (2012) extended a variable neighborhood search (VNS) solution approach for the location routing problem from Pirkwieser and Raidl (2010) and applied it to several instance sets, including the two aforementioned ones with a single depot.

For further details on both problem classes, we refer to the recent survey by Cuda et al. (2015). The previous literature review shows that, on the side of exact methods, the best approaches still cannot consistently solve (in practicable time) 2EVRP instances with more than 50 customers to optimality. On the side of heuristics, only few methods have been designed and tested to deal with instances with more than 50 delivery locations. To the best of our knowledge, only one heuristic has reported, to this date, computational results on larger 2EVRP instances (Hemmelmayr et al., 2012). Hence, there is a need for a more fine-grained study of solution methods for larger problems, as well as for simpler approaches able to efficiently deal with the two families of decisions related to routing and intermediate facilities selection. The proposed method has been designed to cope with these challenges. We developed a technique which performs very well on the classic benchmark instances and, in the meantime, uses fewer and simpler neighbourhood structures than previously published algorithms. During our research, we finally found inconsistencies regarding different benchmark instances used in previous papers. Some slightly different instances have also been referenced with the same name. Thus we collected the different versions and make them available online with unique names in a uniform file format, as described in Section 2.6.1.

2.4 Mathematical model

Different mathematical formulations have been proposed for the 2EVRP (Perboli et al., 2011; Jepsen et al., 2012; Baldacci et al., 2013; Santos et al., 2013, 2015) and for the 2ELRP (Boccia et al., 2011; Contardo et al., 2012). In this section, we display compact formulations based on the model of Cuda et al. (2015).

The 2EVRP can be defined on a weighted undirected graph $G = (N, E)$, where the set of vertices N consists of the depot $\{0\}$, the set of possible satellite locations $S = \{1, \dots, |S|\}$ and the set of customers $C = \{|S|+1, \dots, |S|+|C|\}$. The set of edges E is divided into two subsets, representing the first and second echelon respectively. Set $E^1 = \{(i,j) : i < j,\ i,j \in \{0\} \cup S\}$ represents the edges which can be traversed by first-level vehicles: those connecting the depot to the satellites, and those interconnecting satellites with each other. The set of edges $E^2 = \{(i,j) : i < j,\ i,j \in S \cup C,\ (i,j) \notin S \times S\}$ is used for the second level, and corresponds to possible trips between satellite and customers or pairs of customers.

A fleet of v^1 homogeneous trucks with capacity Q^1 is located at the depot. A total of v^2 homogeneous city freighters are available, each with a given capacity of Q^2. They can be located at any satellite $s \in S$. Still, the number of city freighters at one satellite is limited to v^2_s.

The set R^1 contains all possible routes starting from the depot and delivering a given sequence of customers, then returning to the depot again. Similarly each route r in the set of secondary routes R^2 starts at a satellite $s \in S$, visits one or several customers in C, and returns again to satellite s. Each customer $c \in C$ has a demand of d_c units. Each unit of freight shipped through a satellite induces a handling cost h_s.

Given a secondary route $r \in R^2$ and a customer $c \in C$, the parameter $\beta_{rc} \in \{0, 1\}$ is equal to 1 if and only if customer c is visited in route r, and 0 otherwise. Let $d_r = \sum_{c \in C : c \in r} d_c \leq Q^2$ denote the total demand of customers visited in route r, and p_r represents the cost of each route $r \in R^1 \cup R^2$. The binary variables $x_r \in \{0, 1\}$ with $r \in R^1 \cup R^2$ take the value 1 if and only if route r is in the solution. Finally, each decision variable $q_{rs} \geq 0$ with $r \in R^1,\ s \in S \cap r$, gives the load on the truck on route r that has to be delivered to satellite s.

$$\min \sum_{r \in R^1 \cup R^2} p_r x_r + \sum_{s \in S} h_s \sum_{r \in R^1} q_{rs} \tag{1}$$

s.t.

$$\sum_{r \in R^1} x_r \leq v^1 \tag{2}$$

$$\sum_{r \in R^2} x_r \leq v^2 \tag{3}$$

$$\sum_{r \in R^2 : s \in r} x_r \leq v_s^2 \qquad s \in S \tag{4}$$

$$\sum_{s \in S \cap r} q_{rs} \leq Q^1 x_r \qquad r \in R^1 \tag{5}$$

$$\sum_{r \in R^1} q_{rs} = \sum_{r \in R^2 : s \in r} d_r x_r \qquad s \in S \tag{6}$$

$$\sum_{r \in R^2} \beta_{rc} x_r = 1 \qquad c \in C \tag{7}$$

$$x_r \in \{0, 1\} \qquad r \in R^1 \cup R^2 \tag{8}$$

$$q_{rs} \geq 0 \qquad r \in R^1, s \in S \cap r \tag{9}$$

The objective function (1) sums up routing costs for all routes on both levels and handling costs per unit moved through each satellite. Constraints (2) and (3) set the number of available vehicles for trucks and city freighters, respectively. The number of city freighters per satellite is constrained by Constraint (4). Constraints (5) ensure that the maximum capacity of the trucks is not exceeded. Constraints (6) link the quantities of goods between the first and the second level. They guarantee that the incoming goods equal the outgoing goods at the satellites. As there are no split deliveries allowed on the second level, Constraints (7) ensure that each customer is visited exactly once. The domains of the decision variables are defined by Constraints (8) and (9).

The mathematical model for the 2ELRPSD is similar to the previous model, but needs some adjustments. Each satellite $s \in S$ has a given opening cost f_s and a capacity of k_s units of freight. There is an unbounded number of vehicles available at both levels. Therefore, Constraints (2) to (4) are not needed. The routing of a vehicle incurs fixed costs of f^1 for each truck, and f^2 for each used city freighter. Each binary parameter α_{rs} is equal to 1 if and only if satellite s is visited on route r, and 0 otherwise. If satellite s is opened in the solution, binary variable y_s takes value 1, and 0 otherwise.

Using Constraints (5) to (9), the objective function needs to be changed to $\min \sum_{r \in R^1} (f^1 + p_r) x_r + \sum_{r \in R^2} (f^2 + p_r) x_r + \sum_{s \in S} f_s y_s$ to consider fixed and mileage-based vehicle costs for both levels separately, as well as opening costs for satellites. The capacity limit at the satellites is imposed by $\sum_{r \in R^1} q_{rs} \leq k_s y_s$ for all $s \in S$. If a satellite s has been selected to be open, then the delivery by exactly one truck is guaranteed by Constraints $\sum_{r \in R^1} \alpha_{rs} x_r = y_s$ for all $s \in S$.

2.5 Solution method

The proposed metaheuristic follows the basic structure of a large neighborhood search (LNS), which was first introduced by Shaw (1998). An initial feasible solution is iteratively destroyed and repaired in order to gradually improve the solution. Such a ruin and recreate approach (Schrimpf et al., 2000) has been successfully applied to multiple variants of vehicle routing problems in the past (see, e.g., Pisinger and Ropke 2010). The destruction of parts of a previous solution (ruin) gives freedom to create a new and better solution (recreate). Algorithm 1 shows the basic structure of the proposed method.

Algorithm 1: LNS-2E

```
1  $\mathcal{S}^{best} \leftarrow \mathcal{S} \leftarrow localSearch(repair(instance))$        /* initial solution */
2  $g \leftarrow 0$
3  repeat
4      for $i \leftarrow 0$ to $i_{max}$ do
5          $\mathcal{S}^{temp} \leftarrow localSearch(repair(destroy(\mathcal{S}, g)))$
6          if Satellite was opened/closed during previous destroy phase then
7              $g \leftarrow 0$                                          /* reset grace period */
8          if $cost(\mathcal{S}^{temp}) < cost(\mathcal{S})$ then
9              $\mathcal{S} \leftarrow \mathcal{S}^{temp}$                                  /* accept better solution */
10             $i \leftarrow 0$                                          /* reset re-start period */
11         $g \leftarrow g + 1$
12     if $cost(\mathcal{S}) < cost(\mathcal{S}^{best})$ then
13         $\mathcal{S}^{best} \leftarrow \mathcal{S}$                                      /* store best solution */
14     else
15         $\mathcal{S} \leftarrow localSearch(repair(instance))$          /* re-start: new solution */
16 until $time > time_{max}$
17 return $\mathcal{S}^{best}$
```

At each iteration of the proposed method, 1) a partial solution destruction is performed on the routes of the second level; 2) then the second level is repaired and improved by means of local search, and finally 3) the first level is reconstructed with a simple heuristic. As such, the first level is constructed from scratch in every iteration, but since the number of nodes in the first-level sub-problem is relatively small, this simple heuristic already finds an optimal or near-optimal solution.

Each of the *destroy* phases performs all the destroy operators sequentially as they are described in Section 2.5.1. One single *repair* mechanism is used for solution reconstruction and also to obtain an initial solution (Line 1 and 5). This procedure is described in Section 2.5.2. Afterwards, as needed quantities at the satellites are known, the first level is reconstructed as described in Section 2.5.3.

The choices of intermediate facilities may change as a consequence of the repair operator, or through dedicated destroy operators which temporarily close or re-open some possible locations for intermediate facilities. If a change in open or closed satellites has recently taken place, the status of another satellite will not be changed (Line 5) for a number of iterations that we call *grace period* (g is reset to 0 in Line 7).

We then put emphasis on a strong local search phase, exploiting well-known procedures like 2-opt (Croes, 1958), 2-opt*, or simple relocate and swap moves. *Relocate* shifts a node before or after one of the closest neighbours, if costs are improved. *Swap* explores the exchange of one node with one of the neighbour nodes, as well as exchanging one node with two successive neighbour nodes. This local search phase is applied after the destroy and repair operators. In order to reduce the complexity, moves are only attempted between close customers, as done in the granular search by Toth and Vigo (2003). Further information on these moves can be found in the survey by Vidal et al. (2013).

If a better solution is obtained, it is accepted as the new incumbent solution (Line 9). If no improvement can be found for a large number of iterations, then the algorithm will restart from a new initial solution, even if the objective value is worse (Line 15).

In general, our algorithm requires less local and large neighbourhood operators than the ALNS proposed by Hemmelmayr et al. (2012). The destruction operator parameters are also selected randomly, since the method performed equally well, during our computational experiments, without need for a more advanced adaptive scoring system. In the following, we describe the sets of destroy and repair operators, as well as the management of the decisions related to the first level.

2.5.1 Destroy operators

Our algorithm relies on different destroy operators which are all invoked at each iteration in sequential order. They are applied only to the second level. All of them, except the *open all satellites* neighbourhood, select nodes which are removed from the current solution.

The first four destroy operators are used at each iteration. The last two ones, which change the status of a satellite to *closed* or *open* again, are only invoked if g in Line 5 of Algorithm 1 has exceeded the grace period g_{max} (i.e. no change in open/closed satellites has taken place recently). The destroy operators are now described, in their order of use. When applicable, all random samples are uniformly distributed within their given interval.

Related node removal A seed customer is randomly chosen. A random number of its Euclidean closest customers as well as the seed customer are removed from the current solution and added to the list of nodes to re-insert. This operator receives a parameter p_1, which denotes the maximum percentage of nodes to remove. At most $\lceil p_1 \cdot |C| \rceil$ nodes are removed, with $|C|$ being the overall number of customers.

Biased node removal First, the removal cost of each customer is computed: the savings associated to a removal of node j, located between i and k, is given by $\delta_j = c_{ik} - c_{ij} - c_{jk}$, where c_{ij} denotes the travel cost from node i to node j. The probability of selection of a node for removal is then linearly correlated with the delta evaluation value. The higher the gain after removal, the more likely it will be selected and removed. In every destroy phase, a random percentage of customers from the interval $[0, p_2]$ is removed.

Random route removal Randomly selects routes and removes all containing customers, adding them to the list of nodes to re-insert. This operator randomly selects a number of routes in the interval $[0, \lceil p_3 \cdot \sum_{c \in C} \frac{d_c}{Q^2} \rceil]$.

Remove single node routes This operator removes all routes which contain only one single customer. In the case of the 2EVRP there is a limited number of overall vehicles available, and thus removing the short routes allows to use a vehicle originating from a different satellite in the next repair phase. This operator is used with a probability of $\hat{p}_4$.

The last two destroy operators can be used at most at each g_{max} iterations. During this grace period after satellite selection has been actively altered, none of these two operators will be executed.

Close satellite Chooses a random satellite. If the satellite can be closed and the remaining open ones still can provide sufficient capacity for a feasible solution, the chosen satellite is closed temporarily. All the customers, which are assigned to it, are removed and added to the list of nodes to re-insert. The satellite stays closed until it is opened again in a later phase. This operation is chosen with a probability of $\hat{p}_5$, given variable g has already exceeded the grace period. If this operator has been executed, g is reset to 0.

Open all satellites This neighbourhood makes all previously closed satellites available again. It comes into effect with a probability $\frac{\hat{p}_5}{|S|}$, and thus it depends on the same parameter as the *close satellite* operator and the number of satellites. This operator can only be executed if $g > g_{max}$, i.e. outside the grace period. Its execution resets g to 0.

2.5.2 Repair operator, randomisation and initial solution

At each repair phase, the insertion of the nodes is done in random order. This repair mechanism can sometimes fail, if one customer remains with a higher demand than the largest free capacity available on any vehicle. In this exceptional case, the repair process is restarted, and the nodes are ordered by decreasing demands and then inserted to preserve feasibility.

Repair is achieved with a simplified cheapest insertion heuristic. All nodes are sequentially inserted at their cheapest possible position in the solution. The main difference with the classic cheapest insertion heuristic is that the method does not aim to insert the node with the lowest increase in total costs, but just takes the next candidate from the list and inserts it, in order to reduce complexity and enhance solution diversity. It is a simple and greedy heuristic.

Both the initial solution and every partial solution are always repaired by the same operator. The initial solution can be seen as a "completely destroyed" solution.

After the second level has been repaired, the local search procedure is performed: 2-opt on each of the routes, 2-opt* on all routes originating at the same satellite. The algorithm then tries to relocate single nodes, swap one node with another and to swap two nodes with one other, within a limited neighbourhood of the τ closest nodes, again accepting only improvements. This procedure stops when no improving move exists in the entire neighbourhood. After this procedure, the delivery quantity of each of the satellites is known, and the first level can be constructed using the same insertion heuristic as for the second level and performing local search.

2.5.3 Reconstruction of the first level

For the 2EVRP, it is essential to allow satellites to be delivered by several trucks. In particular, if the demanded quantity at a satellite is larger than a full truckload and no other satellite is available, then there would be no feasible solution. To reconstruct a first level solution, we propose a very simple preprocessing step. Any satellite with a demand larger than a full truckload is virtually duplicated into nodes with demands equal to a truckload, until the remaining demand is smaller than Q^1. The same insertion procedure as the repair operator is used to generate a first level solution. This creates back-and-forth trips to the virtual nodes with demands equal to a full truckload, and completes the solution analogously for the remaining nodes.

Usually there are few nodes associated with the SDVRP on the first level. The largest benchmark instances from literature so far contain only ten satellites at most. This very simple policy enabled to find nearly-optimal first level solutions for most considered instances with limited computational effort. Finally, note that in the considered 2ELRPSD instances, the capacity of a satellite is never larger than the trucks' capacity. Therefore, split deliveries are not generated during reconstruction.

2.6 Computational Experiments

This section describes the currently available sets of instances for the 2EVRP (in Section 2.6.1) and the used instances for the 2ELRPSD (Section 2.6.2) and attempts to resolve some inconsistencies. The calibration of the method is described in Section 2.6.3. The computational results and the comparisons with other state-of-the-art algorithms are discussed in Sections 2.6.4 and 2.6.5. Finally, Section 2.6.6 analyses the sensitivity of the method with respect to several key parameters and design choices.

2.6.1 Benchmark Instances for the two-echelon vehicle routing problem

When looking at the literature, it may appear that there are six unique sets of benchmark instances. However, due to inconsistencies with respect to constraints, nomenclature or locations, we identified in fact several different subsets. In what follows, we explain the differences and provide high quality solutions for them. We consider five different sets of benchmark instances from literature. Sets 2 and 3 were proposed by Perboli et al. (2011) and have been generated based on the instances for the CVRP by Christofides and Eilon. Different customers were chosen and converted into satellites. They also proposed the small Set 1 instances, with just twelve customers and two satellites, which we did not consider. Set 4 was proposed by Crainic et al. (2010); all of them were downloaded from OR-Library (Beasley, 2014).

The instance Sets 2 to 5 as used in Hemmelmayr et al. (2012) were also communicated to us by email (Hemmelmayr, 2013). We noticed a few key differences with the ones available from Beasley (2014).

Set 6 instances were provided from the authors (Baldacci, 2013).

All distances are Euclidean, and computed with double precision. Note that handling costs are set to 0 for all sets except 6b. We will now explain the characteristics of these instance sets in detail, and propose unique names for the sets to overcome existing inconsistencies:

Set 2 There are two different versions in circulation: Please note that the instances with 50 customers in the OR-Library contain a mistake[1]. This can be resolved by exchanging Q^1 and Q^2 capacity values, which is also the way we treated them, like previous authors did.

The names of the instances downloaded from Beasley (2014) and used by Hemmelmayr et al. (2012) were the same, but the instances with 50 customers included different locations for the satellites. For future reference we provide both versions, and we rename the instances with less than 50 customers to Set 2a, the Hemmelmayr (2013) version of 50 customer Set 2 instance files to Set 2b, and the OR-Library version will be called Set 2c. Table 3 shows the characteristics of all Set 2 instances.

Instance names used by Baldacci et al. (2013) have the satellite numbers incremented by one. Apart from that, they are identical with what we received from Hemmelmayr (2013). For example, Set 2a instance named E-n51-k5-s**2-17** (Satellites 2 and 17) corresponds to E-n51-k5-s**3-18** in the result tables of Baldacci et al. (2013).

We provide both versions (OR-Library with corrected capacities as well as the ones received by Hemmelmayr) with distinguishable names at `https://www.univie.ac.at/prolog/research/TwoEVRP`.

Set 3 There are also two different versions of the Set 3 instances in circulation. We collected and solved all of them and distinguished between different versions and identified inconsistencies. Again, the sources Beasley (2014) and Hemmelmayr (2013) were identical for instances with 21 and 32 customers, but different for instances with 50 customers. In the case of Set 3, the filenames for the different instances were also different, so there is no need to introduce new distinguishable names.

The only difference between Set 3 instances with 50 customers from the two sources is the location of the depot. The locations of satellites and customers, as well as the vehicles and demands are identical. All Set 3 instances from Hemmelmayr (2013) place the depot at coordinates (0,0), whereas the files of Beasley (2014) have the depot located at (30,40). Table 13 shows which instances correspond to each other, apart from satellite location.

Please also note that like in Set 2, the Set 3 instances with 50 customers from the OR Library also have the capacities of the two vehicle types interchanged (see Footnote 1). This has been corrected in the files which we provide online.

[1] First level vehicles have a capacity of $Q^1 = 160$ units, and second level vehicles, which by design are supposed to be smaller than level 1 trucks, have a capacity of $Q^2 = 400$ units. For instances with two satellites for example, there are 3 trucks available. They can ship a maximum of $3 * 160 = 480$ units. Overall customers' demand $\sum_{c \in C} d_c$ is larger than 480 units, so there is per se no feasible solution for those instances.

For easier referencing, we also divide the instances in three parts. Set 3a includes all instances with less than 50 customers, Set 3b the larger instances which have been used by Hemmelmayr et al. (2012), and Set 3c the larger instances as they are available at the OR-Library, and have been used by Baldacci et al. (2013), among others.

Set 4 These instances have been treated differently in literature, either with a limit on the number of second level vehicles allowed *per satellite* or only considering a total number of vehicles, with no limitations on the distribution amongst satellites. As proposed in Baldacci et al. (2013), we solved both versions and follow their nomenclature: Set 4a with the limit per satellite, and Set 4b when the constraint of vehicles per satellite is relaxed.

Set 5 This set of instances has been proposed by Hemmelmayr et al. (2012). To the best of our knowledge they were the only ones to report solutions on all instances of that set. Baldacci et al. (2013) were able to find solutions on the small instances with only five satellites.

Set 6 To the best of our knowledge, solutions on these instances have only been reported in Baldacci et al. (2013). Set 6 includes two subsets: Set 6a, with $h_s = 0$, and Set 6b, which considers different handling costs per freight unit at each of the satellites.

Table 1 displays an overview of the characteristics of the individual sets. It lists the number of instances in the according set and subset with number of customers (C), satellites (S), trucks (T), city freighters (CF) and available city freighters per satellite v_s^2. Column hC shows if the handling costs are non-zero. The source of the instance sets is also provided (Hemmelmayr, 2013; Beasley, 2014; Baldacci, 2013).

Table 1: Characteristics and Sources of Instance Sets

Set	Subset	Inst.	C	S	T	CF	hC	v_s^2	HCC	OR-Library	Baldacci
2	a	6	21	2	3	4		-	●	●	
		6	32	2	3	4		-	●	●	
	b	6	50	2	3	5		-	●		
		3	50	4	4	5		-	●		
	c	6	50	2	3	5		-		●	
		3	50	4	4	5		-		●	
3	a	6	21	2	3	4		-	●	●	
		6	32	2	3	4		-	●	●	
	b	6	50	2	3	5		-	●		
	c	6	50	2	3	5		-		●	
4	a	18	50	2	3	6		4		●	
		18	50	3	3	6		3		●	
		18	50	5	3	6		2		●	
	b	18	50	2	3	6		-	●		
		18	50	3	3	6		-	●		
		18	50	5	3	6		-	●		
5		6	100	5	5	[15,32]		-	●		
		6	200	10	5	[17,35]		-	●		
		6	200	10	5	[30,63]		-	●		
6	a	9	50	[4,6]	2	50		-			●
		9	75	[4,6]	3	75		-			●
		9	100	[4,6]	4	100		-			●
	b	9	50	[4,6]	2	50	●	-			●
		9	75	[4,6]	3	75	●	-			●
		9	100	[4,6]	4	100	●	-			●

2.6.2 Benchmark Instances for the two-echelon location routing problem

The proposed algorithm was originally designed for the 2EVRP, nevertheless we also tested it on benchmark instances for the 2ELRPSD. Two sets, called “Nguyen” and “Prodhon” are available at `http://prodhonc.free.fr/Instances/instances0_us.htm`. They present some small errors or unclear descriptions, which are documented in Appendix 2.8.3.

2.6.3 Parameters

The parameters of the proposed method have been calibrated using meta-calibration: the problem of finding good parameters is assimilated to a black-box optimisation problem, in which the method parameters are the decision variables, and the objective function is simulated by running the method on a set of training instances, containing five randomly selected instances, for each set. To perform a fast optimisation we rely on the covariance matrix adaptation evolution strategy (CMA-ES) by Hansen (2006). The source code (in Java) is available at `https://www.lri.fr/~hansen/cmaes_inmatlab.html`.

The performance of our algorithm is rather insensitive to changes in parameters for the small instances, but the rules for closing and opening satellites have to be adjusted to the number of overall available satellites. Our calibration experiments have been conduced for each instance set, independently, and then we searched for one compromise setting for the parameters that yields satisfying results for all different benchmark instances. The calibration results are displayed in Table 2, as well as the average value, standard deviation, and the compromise value which was used for the runs reported in Section 2.6.4.

The size of the limited neighbourhood for the local search relocate and swap moves was also determined by CMA-ES. This parameter always converged to $\tau = 25$ already in early stages of the tuning process, and thus relocate and swap moves are attempted only for nodes within the radius including the 25 Euclidean closest nodes.

Table 2: Parameter values obtained by meta-calibration

	2EVRP							2ELRPSD				
	Set 2	Set 3	Set 4a	Set 4b	Set 5	Set 6a	Set 6b	Nguyen	Prodhon	Mean	Std. Dev.	compromise
p_1	0.35	0.39	0.32	0.31	0.29	0.33	0.26	0.14	0.34	0.30	0.07	0.20
p_2	0.18	0.20	0.12	0.07	0.80	0.52	0.78	0.50	0.19	0.37	0.27	0.35
p_3	0.09	0.07	0.14	0.16	0.14	0.21	0.19	0.28	0.17	0.16	0.06	0.25
$\hat{p}_4$	0.33	0.73	0.19	0.09	0.32	0.21	0.41	0.37	0.57	0.36	0.18	0.50
$\hat{p}_5$	0.06	0.01	0.29	0.24	0.14	0.28	0.26	0.21	0.20	0.19	0.09	0.20

2.6.4 Computational Results

As done in previous literature, we performed five independent runs on each of the 2EVRP benchmark instances and 20 runs on the 2ELRPSD instances. The code is written in Java with JDK 1.7.0_51 and tested on an Intel E5-2670v2 CPU at 2.5 GHz with 3 GB RAM. The code was executed single threaded on one core. We compare the performance of our method on the 2EVRP instances with the hybrid GRASP +VND by Zeng et al. (2014) and the ALNS by Hemmelmayr et al. (2012), when applicable; as well as the currently best known solutions for each instance from the literature. We also show the results of the algorithm on the 2ELRP with single depot and compare with the VNS of Schwengerer et al. (2012). We describe the data of the following tables in general and discuss results in detail on each of the instance sets separately.

Tables 3 to 11 show the characteristics and detailed results for each instance. The columns *C*, *S*, *T* and *CF* display the main characteristics of the instance, where *C* is the number of customers, *S* is the number of satellites, *T* and *CF* the number of available trucks and city freighters, respectively. The last two columns are not applicable for Tables 10 to 11, as they correspond to 2ELRPSD instances with unbounded fleet size.

The next columns display the results of the proposed method (LNS-2E), and methods by Hemmelmayr et al. (2012) (HCC), Zeng et al. (2014) (ZXXS) for the 2EVRP when applicable, and Schwengerer et al. (2012) (SPR) for the 2ELRPSD. The average objective value of five runs is given in column *Avg. 5*. Column *Best 5* shows the best solution found within these five runs, and *Best* gives the best objective value found during all experiments, including parameter calibration. Following the work of Schwengerer et al. (2012), we also used average and best of 20 for the 2ELRPSD for better comparison.

Column *t* reports the average overall runtime of the algorithms in seconds, and *t** the average time when the best solution was found. For easier comparison we chose a simple time limit for termination of our algorithm: 60 seconds for instances with up to 50 customers, and 900 seconds for larger ones. Our time measure corresponds to the wall-clock time of the whole execution of the program, including input and output, computation of the distance matrix, and other pre-processing tasks. Hemmelmayr et al. (2012) and Schwengerer et al. (2012) report

CPU times (which may be slightly smaller than wall clock times). Zeng et al. (2014) only report the time when the best solution was found, but no overall runtime of the algorithm.

BKS refers to the best known solution of that instance. Best known solutions are highlighted in boldface when found by the algorithm, and new BKS are also underlined. We highlight an instance with an asterisk after BKS if the best known solution of the instance is known to be optimal from previous literature.

Tables 3 and 4 provide detailed results on the instances of Set 2 and Set 3. HCC, ZXXS and our algorithm find the best known solutions at every run. The solutions have been proven to be optimal for all the instances except Set 2c and Set 3b. To the best of our knowledge, we are the first ones to report solutions on the 2c instances obtained from Beasley (2014). For Set 3c, the optimal objective values are derived from Baldacci et al. (2013) and Jepsen et al. (2012), but no results from HCC or ZXXS are available. Summarising Tables 3 and 4, we can conclude that Sets 2 and 3 are easy in the sense that all runs of all algorithms always found the optimal or best known solution.

The instances of Set 4 have been addressed in various ways in the literature. Jepsen et al. (2012) considered a limit on the number of city freighters available at each satellite, HCC and ZXXS did not impose this limit, and instead considered the limit on the total number of city freighters only. Baldacci et al. (2013) addressed both variants of the instances to compare their results to both previous results, introducing a new nomenclature: Set 4a for the instances including the limit of vehicles per satellite, and Set 4b when this limit is relaxed.

Tables 5 and 6 display the results on Set 4a and 4b instances. 102 out of the 108 instances have been solved to optimality by Baldacci et al. (2013). Nevertheless we observed small differences of objective values with our solutions (up to a 0.006% difference). This could be explained by a different rounding convention (we use double precision), or by the small optimality gap of Cplex. As a consequence, bold fonts were used for BKS within 0.006% precision. In all these cases the underlying solution is identical, just the objective value is marginally different. We can see from Tables 5 and 6 that in all cases our best solution corresponds to the optimal or best known solution. Only in 3 and 2 instances of Set 4a and 4b, respectively, some of the runs gave slightly worse solutions. On average, in instance Set 4b our results are slightly better than those of the other heuristics.

To the best of our knowledge, Hemmelmayr et al. (2012) were the only authors who published results on the large Set 5 instances with 10 satellites to this date. Baldacci et al. (2013) report solutions on the small Set 5 instances (100 customers/5 satellites), improving three out of six instances to optimality. The algorithm of Hemmelmayr et al. (2012) was evaluated with a limit of 500 iterations. We compare our results in Table 7 and were able to improve the best known solutions on 9 of the 18 instances, depicted with an underlined BKS value. Known optimal solutions are retrieved at least once within the five performed test runs.

Tables 8 and 9 report the results for the instances of Set 6a and b. From the 54 instances, all except 13 solutions have been proven to be optimal, and on nine of those remaining LNS-2E was able to find better solutions. Best solutions were found typically after less than three minutes.

Table 3: Results for Set 2 Instances

					HCC			ZXXS		LNS-2E					
Instance	C	S	T	CF	Avg. 5	t(s)	t*(s)	Avg. 5	t*(s)	Avg. 5	Best 5	Best	t(s)	t*(s)	BKS
Set 2a[1,2]															
E-n22-k4-s6-17	21	2	3	4	**417.07**	37	0	**417.07**	0	**417.07**	**417.07**	**417.07**	60	1	417.07*
E-n22-k4-s8-14	21	2	3	4	**384.96**	34	0	**384.96**	0	**384.96**	**384.96**	**384.96**	60	1	384.96*
E-n22-k4-s9-19	21	2	3	4	**470.60**	35	0	**470.60**	0	**470.60**	**470.60**	**470.60**	60	1	470.60*
E-n22-k4-s10-14	21	2	3	4	**371.50**	37	0	**371.50**	0	**371.50**	**371.50**	**371.50**	60	2	371.50*
E-n22-k4-s11-12	21	2	3	4	**427.22**	31	0	**427.22**	0	**427.22**	**427.22**	**427.22**	60	2	427.22*
E-n22-k4-s12-16	21	2	3	4	**392.78**	36	0	**392.78**	0	**392.78**	**392.78**	**392.78**	60	1	392.78*
E-n33-k4-s14-22	32	2	3	4	**779.05**	85	0	**730.16**	0	**779.05**	**779.05**	**779.05**	60	1	779.05*
E-n33-k4-s1-9	32	2	3	4	**730.16**	74	0	**714.63**	0	**730.16**	**730.16**	**730.16**	60	1	730.16*
E-n33-k4-s2-13	32	2	3	4	**714.63**	64	0	**707.48**	0	**714.63**	**714.63**	**714.63**	60	1	714.63*
E-n33-k4-s3-17	32	2	3	4	**707.48**	58	0	**778.74**	1	**707.48**	**707.48**	**707.48**	60	1	707.48*
E-n33-k4-s4-5	32	2	3	4	**778.74**	77	3	**756.85**	0	**778.74**	**778.74**	**778.74**	60	1	778.74*
E-n33-k4-s7-25	32	2	3	4	**756.85**	53	0	**779.05**	0	**756.85**	**756.85**	**756.85**	60	1	756.85*
Avg.					**577.59**	51	0	**577.59**	0	**577.59**	**577.59**	**577.59**	60	1	577.59
Set 2b[1]															
E-n51-k5-s11-19	50	2	3	5	**581.64**	182	6	**597.49**	1	**581.64**	**581.64**	**581.64**	60	1	581.64*
E-n51-k5-s11-19-27-47	50	4	4	5	**527.63**	147	1	**530.76**	0	**527.63**	**527.63**	**527.63**	60	4	527.63*
E-n51-k5-s2-17	50	2	3	5	**597.49**	100	7	**554.81**	1	**597.49**	**597.49**	**597.49**	60	3	597.49*
E-n51-k5-s2-4-17-46	50	4	4	5	**530.76**	154	1	**581.64**	4	**530.76**	**530.76**	**530.76**	60	3	530.76*
E-n51-k5-s27-47	50	2	3	5	**538.22**	136	1	**538.22**	1	**538.22**	**538.22**	**538.22**	60	1	538.22*
E-n51-k5-s32-37	50	2	3	5	**552.28**	141	1	**552.28**	1	**552.28**	**552.28**	**552.28**	60	2	552.28*
E-n51-k5-s4-46	50	2	3	5	**530.76**	173	0	**530.76**	1	**530.76**	**530.76**	**530.76**	60	3	530.76*
E-n51-k5-s6-12	50	2	3	5	**554.81**	149	2	**531.92**	1	**554.81**	**554.81**	**554.81**	60	4	554.81*
E-n51-k5-s6-12-32-37	50	4	4	5	**531.92**	150	0	**527.63**	1	**531.92**	**531.92**	**531.92**	60	2	531.92*
Avg.					**549.50**	148	2	**549.50**	1	**549.50**	**549.50**	**549.50**	60	2	549.50
Set 2c[2]															
E-n51-k5-s11-19	50	2	3	5						**617.42**	**617.42**	**617.42**	60	3	617.42
E-n51-k5-s11-19-27-47	50	4	4	5						**530.76**	**530.76**	**530.76**	60	1	530.76
E-n51-k5-s2-17	50	2	3	5						**601.39**	**601.39**	**601.39**	60	3	601.39
E-n51-k5-s2-4-17-46	50	4	4	5						**601.39**	**601.39**	**601.39**	60	4	601.39
E-n51-k5-s27-47	50	2	3	5						**530.76**	**530.76**	**530.76**	60	3	530.76
E-n51-k5-s32-37	50	2	3	5						**752.59**	**752.59**	**752.59**	60	5	752.59
E-n51-k5-s4-46	50	2	3	5						**702.33**	**702.33**	**702.33**	60	4	702.33
E-n51-k5-s6-12	50	2	3	5						**567.42**	**567.42**	**567.42**	60	5	567.42
E-n51-k5-s6-12-32-37	50	4	4	5						**567.42**	**567.42**	**567.42**	60	6	567.42
Avg.										**607.94**	**607.94**	**607.94**	60	4	607.94

[1] included in Hemmelmayr (2013)
[2] included in Beasley (2014)

Table 4: Results for Set 3 Instances

					HCC			ZXXS		LNS-2E					
Instance	C	S	T	CF	Avg. 5	t(s)	t*(s)	Avg. 5	t*(s)	Avg. 5	Best 5	Best	t(s)	t*(s)	BKS
Set 3a[1,2]															
E-n22-k4-s13-14	21	2	3	4	**526.15**	43	0	**526.15**	0	**526.15**	**526.15**	**526.15**	60	2	526.15*
E-n22-k4-s13-16	21	2	3	4	**521.09**	44	0	**521.09**	0	**521.09**	**521.09**	**521.09**	60	2	521.09*
E-n22-k4-s13-17	21	2	3	4	**496.38**	49	0	**496.38**	0	**496.38**	**496.38**	**496.38**	60	1	496.38*
E-n22-k4-s14-19	21	2	3	4	**498.80**	43	0	**498.80**	0	**498.80**	**498.80**	**498.80**	60	1	498.80*
E-n22-k4-s17-19	21	2	3	4	**512.81**	26	0	**512.81**	0	**512.80**	**512.80**	**512.80**	60	4	512.80*
E-n22-k4-s19-21	21	2	3	4	**520.42**	34	0	**520.42**	0	**520.42**	**520.42**	**520.42**	60	2	520.42*
E-n33-k4-s16-22	32	2	3	4	**672.17**	76	3	**672.17**	0	**672.17**	**672.17**	**672.17**	60	4	672.17*
E-n33-k4-s16-24	32	2	3	4	**666.02**	77	0	**666.02**	0	**666.02**	**666.02**	**666.02**	60	1	666.02*
E-n33-k4-s19-26	32	2	3	4	**680.36**	84	0	**680.36**	0	**680.36**	**680.36**	**680.36**	60	1	680.36*
E-n33-k4-s22-26	32	2	3	4	**680.37**	77	0	**680.37**	0	**680.36**	**680.36**	**680.36**	60	1	680.36*
E-n33-k4-s24-28	32	2	3	4	**670.43**	88	0	**670.43**	0	**670.43**	**670.43**	**670.43**	60	2	670.43*
E-n33-k4-s25-28	32	2	3	4	**650.58**	63	0	**650.58**	0	**650.58**	**650.58**	**650.58**	60	1	650.58*
Avg.					**591.30**	59	0	**591.30**	0	**591.30**	**591.30**	**591.30**	60	2	591.30
Set 3b[1]															
E-n51-k5-s12-18	50	2	3	5	**690.59**	147	4	**690.59**	1	**690.59**	**690.59**	**690.59**	60	8	690.59
E-n51-k5-s12-41	50	2	3	5	**683.05**	133	38	**683.05**	1	**683.05**	**683.05**	**683.05**	60	11	683.05
E-n51-k5-s12-43	50	2	3	5	**710.41**	217	1	**710.41**	1	**710.41**	**710.41**	**710.41**	60	5	710.41
E-n51-k5-s39-41	50	2	3	5	**728.54**	155	18	**728.54**	4	**728.54**	**728.54**	**728.54**	60	7	728.54
E-n51-k5-s40-41	50	2	3	5	**723.75**	154	17	**723.75**	3	**723.75**	**723.75**	**723.75**	60	5	723.75
E-n51-k5-s40-43	50	2	3	5	**752.15**	158	15	**752.15**	9	**752.15**	**752.15**	**752.15**	60	12	752.15
Avg.					**714.75**	161	16	**714.75**	3	**714.75**	**714.75**	**714.75**	60	8	714.75
Set 3c[2]															
E-n51-k5-s13-19	50	2	3	5						**560.73**	**560.73**	**560.73**	60	10	560.73*
E-n51-k5-s13-42	50	2	3	5						**564.45**	**564.45**	**564.45**	60	3	564.45*
E-n51-k5-s13-44	50	2	3	5						**564.45**	**564.45**	**564.45**	60	2	564.45*
E-n51-k5-s40-42	50	2	3	5						**746.31**	**746.31**	**746.31**	60	5	746.31*
E-n51-k5-s41-42	50	2	3	5						**771.56**	**771.56**	**771.56**	60	15	771.56*
E-n51-k5-s41-44	50	2	3	5						**802.91**	**802.91**	**802.91**	60	16	802.91*
Avg.										**668.40**	**668.40**	**668.40**	60	9	668.40

[1] included in Hemmelmayr (2013)

[2] included in Beasley (2014)

Table 5: Results for Set 4a Instances (with constraint on the number of city freighters per satellite)

					LNS-2E					
Inst.	C	S	T	CF	Avg. 5	Best 5	Best	t(s)	t*(s)	BKS
Set 4a										
1	50	2	3	6	**1569.42**	**1569.42**	**1569.42**	60	4	1569.42*
2	50	2	3	6	**1438.32**	**1438.32**	**1438.32**	60	16	1438.33*
3	50	2	3	6	**1570.43**	**1570.43**	**1570.43**	60	8	1570.43*
4	50	2	3	6	**1424.04**	**1424.04**	**1424.04**	60	7	1424.04*
5	50	2	3	6	**2193.52**	**2193.52**	**2193.52**	60	10	2193.52*
6	50	2	3	6	**1279.89**	**1279.89**	**1279.89**	60	0	1279.87*
7	50	2	3	6	**1458.60**	**1458.60**	**1458.60**	60	2	1458.63*
8	50	2	3	6	**1363.76**	**1363.76**	**1363.76**	60	29	1363.74*
9	50	2	3	6	**1450.25**	**1450.25**	**1450.25**	60	5	1450.27*
10	50	2	3	6	**1407.65**	**1407.65**	**1407.65**	60	6	1407.64*
11	50	2	3	6	2052.21	**2047.43**	**2047.43**	60	3	2047.46*
12	50	2	3	6	**1209.46**	**1209.46**	**1209.46**	60	8	1209.42*
13	50	2	3	6	**1481.80**	**1481.80**	**1481.80**	60	7	1481.83*
14	50	2	3	6	**1393.64**	**1393.64**	**1393.64**	60	1	1393.61*
15	50	2	3	6	**1489.92**	**1489.92**	**1489.92**	60	16	1489.94*
16	50	2	3	6	**1389.20**	**1389.20**	**1389.20**	60	2	1389.17*
17	50	2	3	6	**2088.48**	**2088.48**	**2088.48**	60	15	2088.49*
18	50	2	3	6	**1227.68**	**1227.68**	**1227.68**	60	1	1227.61*
19	50	3	3	6	**1564.66**	**1564.66**	**1564.66**	60	3	1564.66*
20	50	3	3	6	**1272.98**	**1272.98**	**1272.98**	60	25	1272.97*
21	50	3	3	6	**1577.82**	**1577.82**	**1577.82**	60	2	1577.82*
22	50	3	3	6	**1281.83**	**1281.83**	**1281.83**	60	3	1281.83*
23	50	3	3	6	**1807.35**	**1807.35**	**1807.35**	60	8	1807.35*
24	50	3	3	6	**1282.69**	**1282.69**	**1282.69**	60	0	1282.68*
25	50	3	3	6	**1522.40**	**1522.40**	**1522.40**	60	4	1522.42*
26	50	3	3	6	**1167.47**	**1167.47**	**1167.47**	60	1	1167.46*
27	50	3	3	6	**1481.56**	**1481.56**	**1481.56**	60	42	1481.57*
28	50	3	3	6	**1210.46**	**1210.46**	**1210.46**	60	3	1210.44*
29	50	3	3	6	1722.06	**1722.00**	**1722.00**	60	31	1722.04
30	50	3	3	6	**1211.63**	**1211.63**	**1211.63**	60	12	1211.59*
31	50	3	3	6	**1490.32**	**1490.32**	**1490.32**	60	7	1490.34
32	50	3	3	6	**1199.05**	**1199.05**	**1199.05**	60	24	1199.00*
33	50	3	3	6	**1508.32**	**1508.32**	**1508.32**	60	14	1508.30
34	50	3	3	6	**1233.96**	**1233.96**	**1233.96**	60	7	1233.92*
35	50	3	3	6	**1718.42**	**1718.42**	**1718.42**	60	41	1718.41
36	50	3	3	6	**1228.95**	**1228.95**	**1228.95**	60	0	1228.89*
37	50	5	3	6	**1528.73**	**1528.73**	**1528.73**	60	25	1528.73*
38	50	5	3	6	**1169.20**	**1169.20**	**1169.20**	60	15	1169.20*
39	50	5	3	6	**1520.92**	**1520.92**	**1520.92**	60	17	1520.92*
40	50	5	3	6	**1199.42**	**1199.42**	**1199.42**	60	2	1199.42*
41	50	5	3	6	**1667.96**	**1667.96**	**1667.96**	60	9	1667.96*
42	50	5	3	6	**1194.54**	**1194.54**	**1194.54**	60	19	1194.54*
43	50	5	3	6	**1439.67**	**1439.67**	**1439.67**	60	14	1439.67*
44	50	5	3	6	**1045.14**	**1045.14**	**1045.14**	60	24	1045.13*
45	50	5	3	6	1451.48	**1450.95**	**1450.95**	60	3	1450.96*
46	50	5	3	6	**1088.79**	**1088.79**	**1088.79**	60	2	1088.77*
47	50	5	3	6	**1587.29**	**1587.29**	**1587.29**	60	15	1587.29*
48	50	5	3	6	**1082.21**	**1082.21**	**1082.21**	60	23	1082.20*
49	50	5	3	6	**1434.88**	**1434.88**	**1434.88**	60	14	1434.88*
50	50	5	3	6	**1083.16**	**1083.16**	**1083.16**	60	23	1083.12*
51	50	5	3	6	**1398.03**	**1398.03**	**1398.03**	60	21	1398.05*
52	50	5	3	6	**1125.69**	**1125.69**	**1125.69**	60	6	1125.67*
53	50	5	3	6	**1567.79**	**1567.79**	**1567.79**	60	21	1567.77*
54	50	5	3	6	**1127.66**	**1127.66**	**1127.66**	60	15	1127.61*
Avg.					1420.05	**1419.95**	**1419.95**	60	12	1419.94

Table 6: Results for Set 4b Instances ($v_s^2 = v^2$)

					HCC				ZXXS			LNS-2E					
Inst.	C	S	T	CF	Avg. 5	Best	t(s)	t*(s)	Avg. 5	Best	t*(s)	Avg. 5	Best 5	Best	t(s)	t*(s)	BKS
Set 4b																	
1	50	2	3	6	**1569.42**	**1569.42**	235	6	**1569.42**	**1569.42**	1	**1569.42**	**1569.42**	**1569.42**	60	15	1569.42*
2	50	2	3	6	1441.02	**1438.33**	155	43	**1438.33**	**1438.33**	40	**1438.32**	**1438.32**	**1438.32**	60	6	1438.33*
3	50	2	3	6	**1570.43**	**1570.43**	183	3	**1570.43**	**1570.43**	1	**1570.43**	**1570.43**	**1570.43**	60	7	1570.43*
4	50	2	3	6	**1424.04**	**1424.04**	130	11	1429.04	**1424.04**	73	**1424.04**	**1424.04**	**1424.04**	60	7	1424.04*
5	50	2	3	6	**2194.11**	**2194.11**	614	63	**2193.52**	**2193.52**	34	**2193.52**	**2193.52**	**2193.52**	60	18	2193.52*
6	50	2	3	6	**1279.87**	**1279.87**	99	2	**1279.87**	**1279.87**	1	**1279.89**	**1279.89**	**1279.89**	60	0	1279.87*
7	50	2	3	6	1458.63	1458.63	169	6	**1408.57**	**1408.57**	16	**1408.58**	**1408.58**	**1408.58**	60	17	1408.57*
8	50	2	3	6	**1360.32**	**1360.32**	205	5	**1360.32**	**1360.32**	4	**1360.32**	**1360.32**	**1360.32**	60	8	1360.32*
9	50	2	3	6	1450.27	1450.27	204	46	**1403.53**	**1403.53**	4	**1403.53**	**1403.53**	**1403.53**	60	11	1403.53*
10	50	2	3	6	**1360.56**	**1360.56**	174	1	**1360.56**	**1360.56**	1	**1360.54**	**1360.54**	**1360.54**	60	1	1360.56*
11	50	2	3	6	2059.88	2059.88	648	101	2059.41	2059.41	4	2054.60	**2047.43**	**2047.43**	60	2	2047.46*
12	50	2	3	6	**1209.42**	**1209.42**	205	44	**1209.42**	**1209.42**	6	**1209.46**	**1209.46**	**1209.46**	60	20	1209.42*
13	50	2	3	6	1481.83	1481.83	220	25	**1450.93**	**1450.93**	2	**1450.95**	**1450.94**	**1450.94**	60	10	1450.93*
14	50	2	3	6	**1393.61**	**1393.61**	189	6	**1393.61**	**1393.61**	1	**1393.64**	**1393.64**	**1393.64**	60	3	1393.61*
15	50	2	3	6	1489.94	1489.94	173	9	**1466.83**	**1466.83**	1	**1466.84**	**1466.84**	**1466.84**	60	2	1466.83*
16	50	2	3	6	**1387.83**	**1387.83**	147	6	**1387.83**	**1387.83**	6	**1387.85**	**1387.85**	**1387.85**	60	12	1387.83*
17	50	2	3	6	**2088.49**	**2088.49**	625	165	**2088.49**	**2088.49**	27	**2088.48**	**2088.48**	**2088.48**	60	13	2088.49*
18	50	2	3	6	**1227.61**	**1227.61**	94	3	**1227.61**	**1227.61**	1	**1227.68**	**1227.68**	**1227.68**	60	5	1227.61*
19	50	3	3	6	**1546.28**	**1546.28**	171	25	**1546.28**	**1546.28**	25	**1546.28**	**1546.28**	**1546.28**	60	34	1546.28*
20	50	3	3	6	**1272.97**	**1272.97**	99	12	**1272.97**	**1272.97**	57	**1272.98**	**1272.98**	**1272.98**	60	11	1272.97*
21	50	3	3	6	**1577.82**	**1577.82**	155	61	**1577.82**	**1577.82**	19	**1577.82**	**1577.82**	**1577.82**	60	16	1577.82*
22	50	3	3	6	**1281.83**	**1281.83**	127	2	**1281.83**	**1281.83**	28	**1281.83**	**1281.83**	**1281.83**	60	4	1281.83*
23	50	3	3	6	**1652.98**	**1652.98**	175	5	**1652.98**	**1652.98**	3	**1652.98**	**1652.98**	**1652.98**	60	4	1652.98*
24	50	3	3	6	**1282.68**	**1282.68**	110	2	**1282.68**	**1282.68**	1	**1282.69**	**1282.69**	**1282.69**	60	1	1282.68*
25	50	3	3	6	1440.84	1440.68	154	53	**1408.57**	**1408.57**	17	**1408.58**	**1408.58**	**1408.58**	60	20	1408.57*
26	50	3	3	6	**1167.46**	**1167.46**	96	0	**1167.46**	**1167.46**	7	**1167.47**	**1167.47**	**1167.47**	60	16	1167.46*
27	50	3	3	6	1447.79	**1444.50**	163	12	1454.63	**1444.51**	39	**1444.49**	**1444.49**	**1444.49**	60	20	1444.50*
28	50	3	3	6	**1210.44**	**1210.44**	143	7	**1210.44**	**1210.44**	2	**1210.46**	**1210.46**	**1210.46**	60	6	1210.44*
29	50	3	3	6	1561.81	1559.82	178	102	1555.56	**1552.66**	30	**1552.66**	**1552.66**	**1552.66**	60	39	1552.66*
30	50	3	3	6	**1211.59**	**1211.59**	132	5	**1211.59**	**1211.59**	1	**1211.63**	**1211.63**	**1211.63**	60	5	1211.59*
31	50	3	3	6	**1440.86**	**1440.86**	144	37	**1441.07**	**1440.86**	29	**1440.85**	**1440.85**	**1440.85**	60	15	1440.86*
32	50	3	3	6	**1199.00**	**1199.00**	102	11	**1199.00**	**1199.00**	10	**1199.05**	**1199.05**	**1199.05**	60	21	1199.00*
33	50	3	3	6	**1478.86**	**1478.86**	159	16	**1478.86**	**1478.86**	14	**1478.87**	**1478.87**	**1478.87**	60	15	1478.86*
34	50	3	3	6	**1233.92**	**1233.92**	93	4	**1233.92**	**1233.92**	11	**1233.96**	**1233.96**	**1233.96**	60	11	1233.92*
35	50	3	3	6	**1570.80**	**1570.72**	182	116	**1570.72**	**1570.72**	4	**1570.73**	**1570.73**	**1570.73**	60	10	1570.72*
36	50	3	3	6	**1228.89**	**1228.89**	123	6	**1228.89**	**1228.89**	2	**1228.95**	**1228.95**	**1228.95**	60	6	1228.89*
37	50	5	3	6	**1528.81**	**1528.73**	143	55	**1528.98**	**1528.73**	43	**1528.73**	**1528.73**	**1528.73**	60	27	1528.73
38	50	5	3	6	**1163.07**	**1163.07**	88	15	**1163.07**	**1163.07**	1	**1163.07**	**1163.07**	**1163.07**	60	17	1163.07*
39	50	5	3	6	**1520.92**	**1520.92**	158	33	**1520.92**	**1520.92**	16	**1520.92**	**1520.92**	**1520.92**	60	25	1520.92*
40	50	5	3	6	1165.24	**1163.04**	84	20	**1163.04**	**1163.04**	7	**1163.04**	**1163.04**	**1163.04**	60	5	1163.04*
41	50	5	3	6	**1652.98**	**1652.98**	150	12	**1652.98**	**1652.98**	9	**1652.98**	**1652.98**	**1652.98**	60	14	1652.98*
42	50	5	3	6	**1190.17**	**1190.17**	95	31	**1190.17**	**1190.17**	71	**1190.17**	**1190.17**	**1190.17**	60	9	1190.17*
43	50	5	3	6	1408.95	**1406.11**	151	60	**1406.11**	**1406.11**	24	1407.09	**1406.10**	**1406.10**	60	13	1406.11*
44	50	5	3	6	**1035.32**	**1035.03**	109	30	**1035.03**	**1035.03**	7	**1035.05**	**1035.05**	**1035.05**	60	42	1035.03*
45	50	5	3	6	1406.43	1403.10	144	104	**1402.03**	**1402.03**	27	**1401.87**	**1401.87**	**1401.87**	60	21	1401.87*
46	50	5	3	6	1058.97	**1058.11**	74	17	**1058.11**	**1058.11**	7	**1058.10**	**1058.10**	**1058.10**	60	16	1058.11*
47	50	5	3	6	1564.41	1559.82	185	103	1557.04	**1552.66**	25	**1552.66**	**1552.66**	**1552.66**	60	24	1552.66*
48	50	5	3	6	**1074.50**	**1074.50**	83	2	**1074.50**	**1074.50**	1	**1074.51**	**1074.51**	**1074.51**	60	4	1074.50*
49	50	5	3	6	**1435.28**	**1434.88**	140	81	**1434.88**	**1434.88**	38	**1434.88**	**1434.88**	**1434.88**	60	9	1434.88
50	50	5	3	6	**1065.25**	**1065.25**	92	16	**1065.25**	**1065.25**	2	**1065.30**	**1065.30**	**1065.30**	60	33	1065.25*
51	50	5	3	6	**1387.72**	**1387.51**	138	46	**1387.51**	**1387.51**	3	**1387.51**	**1387.51**	**1387.51**	60	19	1387.51*
52	50	5	3	6	**1103.76**	**1103.42**	102	47	**1103.42**	**1103.42**	22	**1103.47**	**1103.47**	**1103.47**	60	17	1103.42*
53	50	5	3	6	**1545.73**	**1545.73**	148	37	**1545.73**	**1545.73**	4	**1545.76**	**1545.76**	**1545.76**	60	21	1545.73*
54	50	5	3	6	**1113.62**	**1113.62**	90	2	**1113.62**	**1113.62**	12	**1113.66**	**1113.66**	**1113.66**	60	5	1113.62*
Avg.					1401.39	1400.96	169	32	1397.69	1397.27	16	1397.21	**1397.06**	**1397.06**	60	14	1397.04

Table 7: Results for Set 5 Instances

					HCC				LNS-2E					
Instance	C	S	T	CF	Avg. 5	Best	t(s)	t*(s)	Avg. 5	Best 5	Best	t(s)	t*(s)	BKS
Set 5														
100-5-1	100	5	5	32	1588.73	1565.45	353	116	1566.87	**1564.46**	**1564.46**	900	201	1564.46*
100-5-1b	100	5	5	15	1126.93	1111.34	397	45	1111.93	**1108.62**	**1108.62**	900	176	1108.62
100-5-2	100	5	5	32	1022.29	**1016.32**	406	117	1017.94	**1016.32**	**1016.32**	900	75	1016.32*
100-5-2b	100	5	5	15	789.05	**782.25**	340	170	783.07	**782.25**	**782.25**	900	152	782.25
100-5-3	100	5	5	30	1046.67	**1045.29**	352	80	**1045.29**	**1045.29**	**1045.29**	900	131	1045.29*
100-5-3b	100	5	5	16	828.99	828.99	391	127	**828.54**	**828.54**	**828.54**	900	62	828.54
100-10-1	100	10	5	35	1137.00	1130.23	429	136	1132.11	1125.53	**1124.93**	900	567	1124.93
100-10-1b	100	10	5	18	928.01	916.48	476	262	922.85	**916.25**	**916.25**	900	424	916.25
100-10-2	100	10	5	33	1009.49	**990.58**	356	232	1014.61	1012.14	1002.15	900	471	990.58
100-10-2b	100	10	5	18	773.58	**768.61**	432	157	786.64	781.27	774.11	900	416	768.61
100-10-3	100	10	5	32	1055.28	**1043.25**	415	209	1053.55	1049.77	1048.53	900	105	1043.25
100-10-3b	100	10	5	17	861.88	**850.92**	418	29	858.72	854.90	854.90	900	175	850.92
200-10-1	200	10	5	62	1626.83	1574.12	888	207	1598.46	1580.34	**1556.79**	900	730	1556.79
200-10-1b	200	10	5	30	1239.79	1201.75	692	374	1217.23	1191.59	**1187.62**	900	588	1187.62
200-10-2	200	10	5	63	1416.87	1374.74	1072	496	1406.16	1366.36	**1365.74**	900	534	1365.74
200-10-2b	200	10	5	30	1018.57	1003.75	1058	221	1016.05	1008.46	**1002.85**	900	721	1002.85
200-10-3	200	10	5	63	1808.24	**1787.73**	916	305	1809.44	1797.80	1793.99	900	675	1787.73
200-10-3b	200	10	5	30	1208.38	1200.74	1217	478	1206.85	1202.21	**1197.90**	900	523	1197.90
Avg.					1138.14	1121.81	589	209	1132.02	1124.01	1120.62	900	374	1118.81

Table 8: Results for Set 6a Instances (no handling costs)

					LNS-2E					
Instance	C	S	T	CF	Avg. 5	Best 5	Best	t(s)	t*(s)	BKS
Set 6a										
A-n51-4	50	4	2	50	**652.00**	**652.00**	**652.00**	60	19	652.00*
A-n51-5	50	5	2	50	**663.41**	**663.41**	**663.41**	60	37	663.41*
A-n51-6	50	6	2	50	**662.51**	**662.51**	**662.51**	60	23	662.51*
A-n76-4	75	4	3	75	985.98	**985.95**	**985.95**	900	128	985.95*
A-n76-5	75	5	3	75	981.19	**979.15**	**979.15**	900	286	979.15*
A-n76-6	75	6	3	75	971.65	**970.20**	**970.20**	900	233	970.20*
A-n101-4	100	4	4	100	1194.38	**1194.17**	**1194.17**	900	267	1194.17*
A-n101-5	100	5	4	100	1215.89	1211.40	1211.40	900	414	1211.38*
A-n101-6	100	6	4	100	1161.91	1158.97	**1155.96**	900	154	1155.96
B-n51-4	50	4	2	50	**563.98**	**563.98**	**563.98**	60	6	563.98*
B-n51-5	50	5	2	50	**549.23**	**549.23**	**549.23**	60	55	549.23*
B-n51-6	50	6	2	50	**556.32**	**556.32**	**556.32**	60	39	556.32*
B-n76-4	75	4	3	75	793.97	**792.73**	**792.73**	900	320	792.73*
B-n76-5	75	5	3	75	784.27	784.19	784.19	900	190	783.93*
B-n76-6	75	6	3	75	775.75	774.24	**774.17**	900	160	774.17*
B-n101-4	100	4	4	100	939.79	**939.21**	**939.21**	900	377	939.21*
B-n101-5	100	5	4	100	971.27	969.13	969.13	900	161	967.82*
B-n101-6	100	6	4	100	961.91	**960.29**	**960.29**	900	88	960.29*
C-n51-4	50	4	2	50	**689.18**	**689.18**	**689.18**	60	13	689.18*
C-n51-5	50	5	2	50	**723.12**	**723.12**	**723.12**	60	19	723.12*
C-n51-6	50	6	2	50	**697.00**	**697.00**	**697.00**	60	46	697.00*
C-n76-4	75	4	3	75	1055.61	**1054.89**	**1054.89**	900	339	1054.89*
C-n76-5	75	5	3	75	**1115.32**	**1115.32**	**1115.32**	900	113	1115.32*
C-n76-6	75	6	3	75	1066.88	**1060.52**	**1060.52**	900	474	1060.52
C-n101-4	100	4	4	100	1305.94	**1302.16**	**1302.16**	900	236	1302.16
C-n101-5	100	5	4	100	1307.24	1306.27	**1305.82**	900	141	1305.82
C-n101-6	100	6	4	100	1292.10	**1284.48**	**1284.48**	900	446	1284.48
Avg.					912.51	911.11	910.98	620	177	910.92

Table 9: Results for Set 6b Instances ($h_s \neq 0$)

					LNS-2E					
Instance	C	S	T	CF	Avg. 5	Best 5	Best	t(s)	t*(s)	BKS
Set 6b										
A-n51-4	50	4	2	50	**744.24**	**744.24**	**744.24**	60	17	744.24*
A-n51-5	50	5	2	50	**811.51**	**811.51**	**811.51**	60	49	811.52*
A-n51-6	50	6	2	50	**930.11**	**930.11**	**930.11**	60	31	930.11*
A-n76-4	75	4	3	75	**1385.51**	**1385.51**	**1385.51**	900	26	1385.51*
A-n76-5	75	5	3	75	**1519.86**	**1519.86**	**1519.86**	900	71	1519.86*
A-n76-6	75	6	3	75	1666.28	**1666.06**	**1666.06**	900	533	1666.06*
A-n101-4	100	4	4	100	1884.48	1883.79	1883.79	900	283	1881.44*
A-n101-5	100	5	4	100	1723.06	1714.58	1711.95	900	318	1709.06
A-n101-6	100	6	4	100	1795.36	1793.76	1791.44	900	100	1777.69
B-n51-4	50	4	2	50	**653.09**	**653.09**	**653.09**	60	21	653.09*
B-n51-5	50	5	2	50	**672.10**	**672.10**	**672.10**	60	51	672.10*
B-n51-6	50	6	2	50	**767.13**	**767.13**	**767.13**	60	58	767.13*
B-n76-4	75	4	3	75	**1094.52**	**1094.52**	**1094.52**	900	44	1094.52*
B-n76-5	75	5	3	75	**1218.12**	**1218.12**	**1218.12**	900	19	1218.13*
B-n76-6	75	6	3	75	1328.90	1328.90	**1326.76**	900	329	1326.76*
B-n101-4	100	4	4	100	1500.80	**1500.55**	**1500.55**	900	82	1500.55
B-n101-5	100	5	4	100	1398.05	1398.05	**1395.32**	900	317	1395.32
B-n101-6	100	6	4	100	1455.05	1453.54	1453.54	900	460	1450.39
C-n51-4	50	4	2	50	**866.58**	**866.58**	**866.58**	60	22	866.58*
C-n51-5	50	5	2	50	**943.12**	**943.12**	**943.12**	60	12	943.12*
C-n51-6	50	6	2	50	**1050.42**	**1050.42**	**1050.42**	60	18	1050.42*
C-n76-4	75	4	3	75	1439.39	**1438.96**	**1438.96**	900	84	1438.96*
C-n76-5	75	5	3	75	1745.49	**1745.39**	**1745.39**	900	281	1745.39*
C-n76-6	75	6	3	75	1759.40	**1756.54**	**1756.54**	900	358	1756.54*
C-n101-4	100	4	4	100	2076.36	2073.84	**2064.86**	900	378	2064.86
C-n101-5	100	5	4	100	1974.39	1967.51	**1964.63**	900	285	1964.63
C-n101-6	100	6	4	100	1867.45	**1861.50**	**1861.50**	900	401	1861.50
Avg.					1343.36	1342.20	1341.39	620	172	1340.57

To test the robustness of our algorithm we applied it also to the 2ELRPSD instances without any further changing of operators or tuning. The results can be seen in Tables 10 and 11. BKS are derived from Schwengerer et al. (2012); Nguyen et al. (2012a,b); Contardo et al. (2012). LNS-2E is competitive on this problem class, too, with solutions being on average within 0.6% of the solutions found by the state-of-the-art VNS by Schwengerer et al. (2012) (SPR).

Table 10: Results for 2ELRPSD Instances Set Nguyen

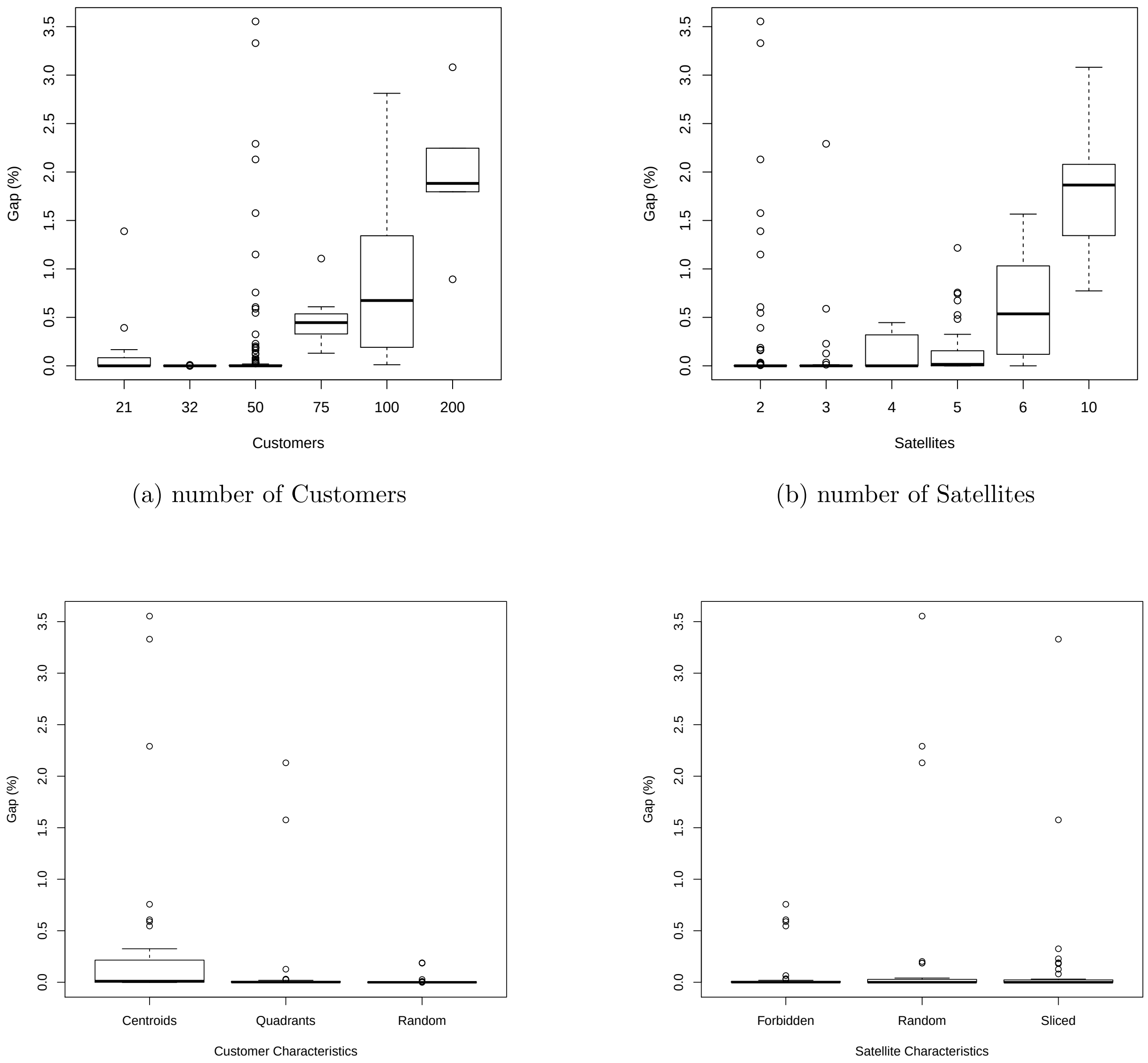
			SPR				LNS-2E					
Instance	C	S	Avg. 20	Best 20	t(s)	t*(s)	Avg. 20	Best 20	Best	t(s)	t*(s)	BKS
Set Nguyen												
25-5N	25	5	**80370.00**	**80370**	76	3	**80370.00**	**80370**	**80370**	60	5	80370
25-5Nb	25	5	**64562.00**	**64562**	91	0	**64562.00**	**64562**	**64562**	60	16	64562
25-5MN	25	5	**78947.00**	**78947**	61	1	**78947.00**	**78947**	**78947**	60	6	78947
25-5MNb	25	5	**64438.00**	**64438**	89	0	**64438.00**	**64438**	**64438**	60	4	64438
50-5N	50	5	**137815.00**	**137815**	116	35	**137815.00**	**137815**	**137815**	60	25	137815
50-5Nb	50	5	110204.40	**110094**	132	51	110981.85	**110094**	**110094**	60	22	110094
50-5MN	50	5	**123484.00**	**123484**	125	41	**123484.00**	**123484**	**123484**	60	4	123484
50-5MNb	50	5	105687.00	**105401**	202	48	105783.45	**105401**	**105401**	60	16	105401
50-10N	50	10	**115725.00**	**115725**	143	24	117325.55	**115725**	**115725**	60	20	115725
50-10Nb	50	10	87345.00	**87315**	176	66	88212.00	87520	**87315**	60	23	87315
50-10MN	50	10	**135519.00**	**135519**	144	10	138241.35	**135519**	**135519**	60	14	135519
50-10MNb	50	10	**110613.00**	**110613**	218	13	111520.80	**110613**	**110613**	60	22	110613
100-5N	100	5	200685.05	**193228**	168	101	193806.85	193229	193229	900	239	193228
100-5Nb	100	5	164508.10	**158927**	258	144	159064.10	**158927**	**158927**	900	315	158927
100-5MN	100	5	206567.40	**204682**	184	159	204876.10	**204682**	**204682**	900	105	204682
100-5MNb	100	5	166357.35	**165744**	315	247	165795.35	**165744**	**165744**	900	252	165744
100-10N	100	10	214585.60	**209952**	223	167	216265.50	210799	**209952**	900	344	209952
100-10Nb	100	10	155790.60	**155489**	352	251	161273.30	**155489**	**155489**	900	442	155489
100-10MN	100	10	203798.05	**201275**	229	163	204396.15	**201275**	**201275**	900	324	201275
100-10MNb	100	10	170791.25	**170625**	347	283	172202.45	**170625**	**170625**	900	268	170625
200-10N	200	10	349584.15	**345267**	641	525	359948.65	350680	350680	900	758	345267
200-10Nb	200	10	264228.90	**256171**	907	791	260698.20	257191	257191	900	748	256171
200-10MN	200	10	332207.50	**323801**	453	441	329486.45	324279	324279	900	777	323801
200-10MNb	200	10	292036.65	**287076**	944	843	297857.50	293339	290702	900	778	287076
Avg.			163993.75	161938	275	184	164472.98	162531	162377	480	230	161938

Table 11: Results for 2ELRPSD Instances Set Prodhon

			SPR				LNS-2E					
Instance	C	S	Avg. 20	Best 20	t(s)	t*(s)	Avg. 20	Best 20	Best	t(s)	t*(s)	BKS
Set Prodhon												
20-5-1	20	5	**89075.00**	**89075**	63	2	**89075.00**	**89075**	**89075**	60	4	89075
20-5-1b	20	5	**61863.00**	**61863**	83	0	**61863.00**	**61863**	**61863**	60	2	61863
20-5-2	20	5	84489.50	**84478**	62	11	**84478.00**	**84478**	**84478**	60	19	84478
20-5-2b	20	5	61033.80	**60838**	125	0	**60838.00**	**60838**	**60838**	60	5	60838
50-5-1	50	5	130859.30	**130843**	80	16	131454.00	131085	**130843**	60	24	130843
50-5-1b	50	5	101548.00	**101530**	128	35	101669.20	**101530**	**101530**	60	14	101530
50-5-2	50	5	**131825.00**	**131825**	97	11	131827.00	**131825**	**131825**	60	16	131825
50-5-2b	50	5	**110332.00**	**110332**	198	12	**110332.00**	**110332**	**110332**	60	14	110332
50-5-2BIS	50	5	**122599.00**	**122599**	112	91	**122599.00**	**122599**	**122599**	60	3	122599
50-5-2bBIS	50	5	105935.50	**105696**	198	155	105707.85	**105696**	**105696**	60	18	105696
50-5-3	50	5	128436.00	**128379**	80	9	128614.50	**128379**	**128379**	60	10	128379
50-5-3b	50	5	**104006.00**	**104006**	131	6	**104006.00**	**104006**	**104006**	60	7	104006
100-5-1	100	5	318667.00	318225	226	153	319268.60	318399	318399	900	303	318134
100-5-1b	100	5	257436.35	256991	301	220	257686.40	256991	256888	900	268	256878
100-5-2	100	5	231340.00	**231305**	204	131	231488.85	**231305**	**231305**	900	229	231305
100-5-2b	100	5	194812.70	194763	240	202	194800.35	194763	194729	900	176	194728
100-5-3	100	5	245334.90	244470	174	124	245178.75	**244071**	**244071**	900	377	244071
100-5-3b	100	5	195586.20	195381	180	111	195123.20	**194110**	**194110**	900	327	194110
100-10-1	100	10	357381.40	352694	233	167	362648.70	354525	352122	900	401	351243
100-10-1b	100	10	300239.15	298186	299	194	312451.60	299758	298298	900	540	297167
100-10-2	100	10	304931.20	304507	248	194	307937.60	304909	**304438**	900	520	304438
100-10-2b	100	10	264592.00	264092	307	208	265814.85	264173	263876	900	310	263873
100-10-3	100	10	312701.25	311447	227	141	318952.10	311699	310930	900	512	310200
100-10-3b	100	10	261577.90	260516	303	218	265442.40	262932	261566	900	437	260328
200-10-1	200	10	552488.90	548730	1009	748	564159.80	550672	550672	900	725	548703
200-10-1b	200	10	448095.45	445791	635	576	456952.40	448188	447113	900	692	445301
200-10-2	200	10	513673.40	**497451**	1158	832	499499.45	498486	498397	900	656	497451
200-10-2b	200	10	432687.00	**422668**	730	696	428912.35	422967	422877	900	601	422668
200-10-3	200	10	529578.00	**527162**	970	903	568539.15	534271	533174	900	668	527162
200-10-3b	200	10	404431.25	402117	592	557	425078.20	417686	417429	900	700	401672
Avg.			245251.87	243599	313	224	248413.28	244720	244395	564	286	243363

Finding high-quality solutions is increasingly difficult as the problem size grows, and different instance characteristics influence solution quality. Figure 2 displays boxplots of 2EVRP instances grouped together by number of customers (2a), or number of satellites (2b), similar customer distribution (2c) and similar satellite distribution (2d) using gaps of the average value of five runs to BKS.

Problem difficulty quickly grows with the number of satellites, as well as with the number of customers, as plotted in the upper part of Figure 2. The number of samples for the different classes varies a lot: There are only 18 instances with 75 customers, but 165 instances with 50 customers, which explains the large number of outliers for those instances. Crainic et al. (2010) provide a detailed overview on the distribution of customers and satellites in instances of Set 4. There are three distribution patterns for customers: *random*, with equally distributed nodes; *centroids*, where more customers are located in six centroids in a central zone, and some customers closer together in four outer areas, representing suburbs. In the *quadrant* pattern customers are arranged in conglomerations of higher density in each of the four quadrants. The three patterns for satellites are: *random*, where satellites are randomly placed on a ring around the customers; *sliced*, with the satellites distributed more evenly on the ring around customers; and *forbidden zone*: a random angle on the ring was chosen where no satellites could be used, to simulate various conditions like cities located near lakes or mountains. Figure 2c shows



(a) number of Customers

(b) number of Satellites

(c) Set 4: Customer distribution

(d) Set 4: Satellite distribution

Figure 2: Boxplots of solution quality for instances grouped by number of customers/satellites and distribution characteristics

The hinges depict approximately the first and third quartile of the solutions. The whiskers extend to $\pm 1.58 \frac{Q_3 - Q_1}{\sqrt{n}}$. (The R Foundation for Statistical Computing, 2014). See the Appendix 2.8.2 for a detailed definition.

that instances with customers located in centroids are harder to solve. On the other hand, the distribution pattern of the satellites doesn't have a large impact on solution quality.

2.6.5 Graphical Example

Several structurally different 2EVRP solutions can have similar objective values. We discuss and visualise this with the help of a demonstrative graphical example. Also, the solution differences for an instance with or without constraining the number of city freighters per satellite are pointed out.

The selection of the correct subset of satellites to use is crucial. A graphical representation of different solutions of the Set 4 instance 38 is provided in Figure 3. The locations of the depot (square), satellites (triangles) and customers (circles) is the same for each solution. Nevertheless, the obtained vehicle routes are substantially different given different subsets of open satellites.

Figure 3a represents the optimal solution to the Set 4a instance 38. A maximum of $v_s^2 = 2$ city freighters per satellite are available. The total demand of all customers sums up to 20206 units of freight. The capacity Q^2 of a city freighter is 5000 units. Any feasible solution needs at least $\lceil \frac{20206}{5000} \rceil = 5$ city freighters, and thus at least $\lceil 5/2 \rceil = 3$ satellites have to be used. Two city freighter routes are located on the left of the figure, and three city freighter routes on the right hand side, where two city freighters leave from the same satellite, and a third one from a close by satellite.

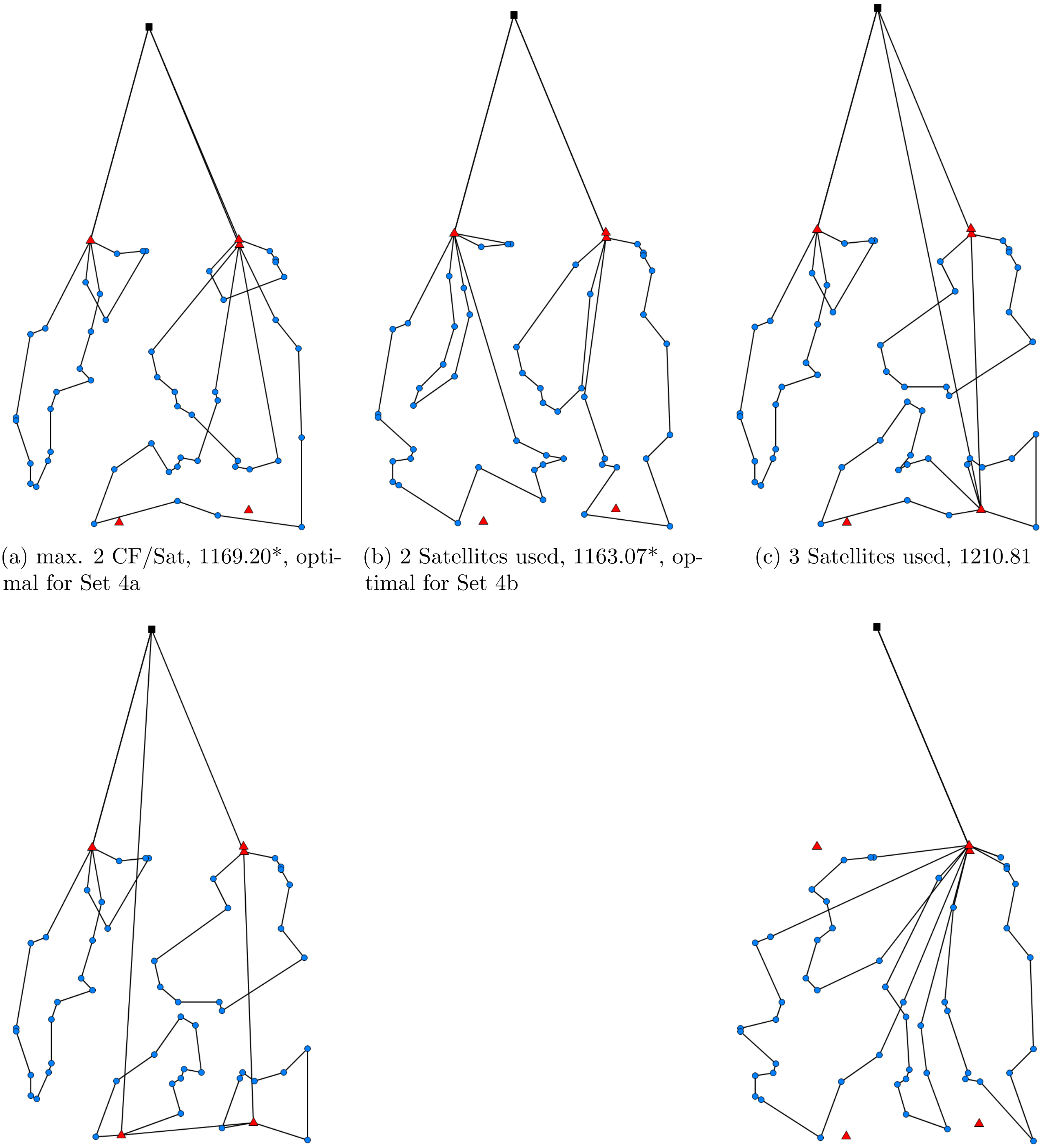
Considering the instance as in Set 4b, with a global number of city freighters available but no constraints on the distribution amongst satellites ($v_s^2 = v^2$), the optimal solution is displayed in Figure 3b. Three city freighters can leave the same satellite, as is the case on the left hand side. Only two of the five satellites have to be used.

LNS-2E starts with the construction of routes at the second level. In early stages of the optimisation process, all customers are likely assigned to their closest satellites. Without neighbourhoods that impact the selection of satellites, the algorithm would likely be trapped in a local optimum such as the one of Figure 3c. If partial routes originating at the bottom right satellite exist, customers may be sequentially inserted in those routes and the bottom left satellite may not be opened. The solution displayed in Figure 3d is often obtained. It has the best cost considering only the second level, but long truck routes on the first level set off this gain, leading to a worse solution overall.

Figure 3e shows the best found solution, if only the top right satellite is open. The cost differences from one solution to the next one are small, although the solution itself is fundamentally different. If the two satellites at the bottom are both selected for closure at the same time, then the algorithm finds the optimal solution within seconds.

We tried different strategies to evaluate the chances of a satellite to be included in the best solution: taking into account a delta evaluation on the truck route, or combining this value with the total units shipped through this satellite; or the absolute distance from the depot. We observed that closing satellites randomly is a straightforward and very simple approach, which performs quite well on average over all the different benchmark instances, whereas other techniques present advantages and disadvantages in several special cases.

For further research, we suggest to shift the cost structure towards more realistic scenarios. In the classic 2EVRP as we considered it, the cost of large trucks and small city freighters is



(a) max. 2 CF/Sat, 1169.20*, optimal for Set 4a

(b) 2 Satellites used, 1163.07*, optimal for Set 4b

(c) 3 Satellites used, 1210.81

(d) 4 Satellites used, 1226.43

(e) 1 Satellite used, 1255.98

Figure 3: Different Solutions of Instance 38 from Set 4, depending on satellite openings and v_s^2

the same. For instance Sets 2 to 4, the capacity of a truck is 2.5 times higher than of a city freighter. For Set 5 instances this ratio gets up to more than 14, so one can safely assume that the operating cost of a truck will be higher than of a city freighter in more realistic set-ups. This has not yet been taken into account in previous publications on the 2EVRP, and would lead to large differences between the solution costs of Figure 3.

2.6.6 Sensitivity Analysis

Sensitivity analyses were conducted to evaluate the impact of major parameters and components of the method. In particular, we evaluate the impact of disabling single destroy operators or local search procedures at a time and provide the average objective value over five runs of all benchmark instances. Table 12 shows the average objective values for instances of each class and its average deviation (Gap (%)) when disabling elements of the algorithm.

The elimination of the open all satellites (*no open*) neighbourhood has a strong impact on solution quality. As closed satellites are only opened again in a re-start phase, the algorithm is likely to be trapped in a local optimum.

If no satellites are forced to be closed, there is no need to open up any satellite again. In this case, solution quality deteriorates by 1.15% on average (*no close*). For some instances, satellites located very far away from the depot will not be used in the best solution, but on the second level it seems to be beneficial to use them for customers close by. The impact on the 2ELRPSD, especially the instance Set Prodhon is quite strong, as the fixed costs of a satellite are not taken into account on the second level. If a satellite with high opening costs is located close to many customers, it will very likely be used, although it would pay off to use a cheaper one further away. A similar behaviour was discussed in Section 2.6.5 and Figure 3.

Some techniques work better on smaller instances, while others perform better on the larger instances. For some cases we even observed small improvements when an operator was not used. Not using biased node removal for example is needed to robustly find optimal solutions on the smaller instance sets (2-4) but yields improvements on the larger sets.

For local search techniques, we also observe differences on methods for small or large instance sets respectively. Removing classic 2-opt has a small impact. The repair mechanism finds already high quality single routes in terms of 2-optimality. Eliminating the inter-tour operator 2-opt* has a stronger effect on larger Set 5 and Set 6 instances, which contain more routes than the smaller instances. The relocate neighbourhood is essential for instances with more than 50 customers, whereas exchanging two nodes against one can have a negative effect on those larger instances. We still decided to keep it in the design of the algorithm, as this local search is needed to find optimal solutions for smaller instances. Of course the algorithm could be fine-tuned for specific applications or instance sizes.

Table 12: Sensitivity Analysis and contribution of individual components

			Sensitivity Analysis				
Set	base	no related	no biased	no route	no single	no close	no open
2	578.27	0.21%	0.02%	0.17%	0.02%	0.07%	2.07%
3	641.44	0.31%	0.06%	0.27%	0.06%	0.07%	1.66%
4a	1420.05	0.25%	0.02%	0.10%	0.04%	0.53%	1.37%
4b	1397.21	0.16%	-0.01%	0.18%	0.05%	1.14%	1.39%
5	1132.02	0.69%	-0.07%	0.31%	0.35%	0.39%	2.35%
6a	912.51	0.25%	-0.03%	0.07%	-0.01%	1.20%	1.36%
6b	1343.36	0.15%	0.04%	0.06%	0.01%	1.15%	0.97%
Prodhon	248413.28	0.95%	-0.31%	-0.55%	-0.36%	4.65%	3.58%
Nguyen	164492.82	0.61%	0.05%	0.28%	0.19%	1.13%	1.38%
Avg.	46703.44	0.40%	-0.03%	0.10%	0.04%	1.15%	1.79%

Set	base	no restart	no 2opt	no 2opt*	no relocate	no swap	no swap 2
2	578.27	0.08%	0.04%	0.10%	0.00%	0.00%	0.00%
3	641.44	0.13%	0.05%	0.09%	0.06%	0.08%	0.18%
4a	1420.05	0.11%	0.05%	0.03%	-0.01%	0.01%	-0.01%
4b	1397.21	0.10%	0.03%	0.27%	0.04%	0.02%	0.03%
5	1132.02	0.06%	0.31%	0.53%	0.47%	0.29%	0.09%
6a	912.51	0.11%	0.02%	0.15%	0.24%	-0.03%	0.02%
6b	1343.36	0.07%	0.05%	0.17%	0.12%	0.05%	-0.01%
Prodhon	248413.28	0.43%	-0.29%	0.30%	0.56%	-0.51%	0.00%
Nguyen	164492.82	-0.08%	-0.20%	0.21%	0.40%	0.22%	0.03%
Avg.	46703.44	0.11%	0.01%	0.21%	0.21%	0.01%	0.04%

2.7 Conclusion

We presented a very simple and fast LNS heuristic for the 2EVRP and 2ELRPSD. LNS-2E makes use of one repair operator and only a few destroy operators. The proposed method finds solutions of higher quality than existing algorithms, while being fast and conceptually simpler. The impact of various parameters and design choices was highlighted. Meta-calibration techniques were used to set the parameters to good values, which were subsequently verified during sensitivity analyses. Different techniques were attempted to open or close satellites, and thus explore various combinations of design choices. At the end, a simple randomised approach for fixing satellites, assorted with a minimum number of iterations without change of this decision led to good and robust results on a wide range of benchmark instances. LNS-2E was able to improve 18 best known solutions on the 49 2EVRP instances for which no proven optimal solution exists so far. Having resolved the inconsistencies on the different sets of benchmark instances used in literature paves the way for future research on this topic, which will focus on solving rich city logistics problems, and shifting the cost structure to a more realistic scenario. It will be interesting to examine the implications of using higher operating costs for larger trucks than for smaller city freighters, approaching more realistic and real-life transportation problems.

Online Resources

All necessary data can be found in the online section at `https://www.univie.ac.at/prolog/research/TwoEVRP`. All instances have been transformed into a uniform format and can be downloaded. Set 4 instances used to have negative and real x/y coordinates (with a maximum of two positions after decimal space). We added a fixed factor to shift them only to be positive and multiplied them by 100 to be able to use positive integers. This does not change the solution, but note that the objective should be adjusted by a factor 100. We also provide detailed results on the new found best known solutions, both in human readable text and a graphical representation.

Acknowledgement

This work is partially funded by the Austrian Climate and Energy Fund within the “Electric Mobility Flagship projects” program under grant 834868 (project VECEPT). The financial support by the Austrian Federal Ministry of Science, Research and Economy and the National Foundation for Research, Technology and Development is gratefully acknowledged.

We would like to thank Roberto Baldacci, Vera Hemmelmayr, Sandro Pirkwieser, Martin Schwengerer and Gianfranco Guastaroba for providing us detailed information, and/or solutions on the instances solved by their methods.

We would also like to thank the two anonymous reviewers for their helpful and constructive comments which helped us to improve the manuscript.

2.8 Appendix

2.8.1 Nomenclature of Set 3 instances

Table 13: Set 3 instances with 50 customers: identical except for depot coordinates

Set 3b[1]		Set 3c[2]	
depot at (0,0)		depot at (30,40)	
E-n51-k5-	12-18	E-n51-k5-	13-19
	12-41		13-42
	12-43		13-44
	39-41		40-42
	40-41		41-42
	40-43		41-44

[1] included in Hemmelmayr (2013)
[2] included in Beasley (2014)

Hemmelmayr et al. (2012) report a best solution with total costs of 690.59 for instance E-n51-k5-s**12-18**, which corresponds exactly to the solution found by our algorithm. On the other hand Jepsen et al. (2012) report an objective value of 560.73 for an instance of that name. This corresponds exactly to the objective value found by our algorithm for the instance

E-n51-k5-s**13-19** from Beasley (2014). So we assume Jepsen et al. (2012) used the same instances as are provided from Beasley (2014), but the IDs of satellites in the names have to be increased by one. Both versions exist in literature, as we described in Section 2.6.1. The only difference between Sets 3b and 3c is the location of the depot. Table 13 shows, which instances correspond to each other, apart from depot coordinates.

2.8.2 Definition of boxplots in Figure 2

The two hinges are versions of the first (Q_1) and third quartile (Q_3), i.e., close to quantile (x, c(1,3)/4). The hinges equal the quartiles for odd n (where $n \leftarrow \; length(x)$) and differ for even n. Whereas the quartiles only equal observations for $n = (1 \mod 4)$, the hinges do so additionally for $n = (2 \mod 4)$, and are in the middle of two observations otherwise. They are based on asymptotic normality of the median and roughly equal sample sizes for the two medians being compared, and are said to be rather insensitive to the underlying distributions of the samples. The notches extend to ± $1.58\frac{IQR}{\sqrt{n}}$, where Interquartile Range $IQR = Q_3 - Q_1$. This gives roughly a 95% confidence interval for the difference in two medians. (The R Foundation for Statistical Computing, 2014)

2.8.3 2ELRPSD instances

The instances for the 2ELRPSD were downloaded at the homepage of Caroline Prodhon: `http://prodhonc.free.fr/Instances/instances0_us.htm`. If the description file is renamed to ".doc" it can be opened in human readable format by Word. We found some small typos in the description: At the end of the Prodhon files first the fixed cost of a second level vehicle (F2) is given, then the fixed cost of a truck on first level (F1).

The description of the cost structure also seems to contain an error: According to the description, second level vehicles operate at higher cost per distance than first level trucks, which should be the other way round, obviously. This is also the way we treated the instances, and we believe previous authors did, too. The cost linking point A to point B was calculated as shown in Table 14. Please note that first level vehicles do not operate at exactly twice the Euclidean distance, due to the use of the ceil function *after* multiplication by factor 2.

Table 14: Distance Matrix Calculation for the 2ELRPSD

	Instance set	
	Prodhon	Nguyen
first level	$\lceil\sqrt{(x_A - x_B)^2 + (y_A - y_B)^2} * 100 * 2\rceil$	$\lceil\sqrt{(x_A - x_B)^2 + (y_A - y_B)^2} * 10 * 2\rceil$
second level	$\lceil\sqrt{(x_A - x_B)^2 + (y_A - y_B)^2} * 100\rceil$	$\lceil\sqrt{(x_A - x_B)^2 + (y_A - y_B)^2} * 10\rceil$

In the instance file 200-10-3b, the capacity of first level vehicles is missing, and thus we used 5000 (as this is the value used for all the other Prodhon instance files). Some customers in the 2ELRPSD instances have a demand of 0. This case was not explicitly dealt with in the papers: our algorithm still plans an itinerary for a city freighter which will visit the customer, but does not deliver any quantity of goods.

3 The electric two-echelon vehicle routing problem

This part is partially based on Breunig et al. (2017) and published as Breunig et al. (2019).

3.1 Introduction

Nowadays, as technology for electric mobility progresses, multi-tier delivery schemes are naturally destined to make use of electric vehicles, and multiple companies have, in practice, already operated this transition (Foltyński, 2014). Yet, electric vehicles also pose specific challenges, due to their limited autonomy, smaller capacity, and the possible need of planned visits to charging stations. Moreover, whereas early charging technologies required several hours for a full recharge, recent developments of fast-charging or battery-swap stations (Yang and Sun, 2015; Hof et al., 2017; Keskin and Çatay, 2018) allow energy replenishment in half an hour. The growing adoption of these technologies allows en-route recharging (e.g., during lunch breaks) in metropolitan distribution systems, as well as the use of cheaper lightweight vehicles (Perboli et al., 2018a; Perboli and Rosano, 2018) with smaller batteries. Last but not the least, the study on battery-powered distribution is not limited to terrestrial vehicles, but also meets critical applications in last-mile distribution using aerial vehicles (i.e., drones – Poikonen et al. 2017 and Wang et al. 2017b), which typically have a smaller autonomy.

To focus on these challenges, we introduce the E2EVRP as a prototypical problem. It is a natural extension of the 2EVRP in which electric vehicles are used on the second echelon. Given a geographically-dispersed set of customers demanding an amount of a single commodity, a set of satellites (intermediate facilities), a set of charging stations, and a central depot where the commodity is kept, the E2EVRP seeks least-cost delivery routes to transport the commodity from the depot to the satellites with conventional vehicles (first-level), and from the satellites to the customers using an electric fleet (second-level). Some additional rules must be satisfied with respect to the basic 2EVRP:

- Electric vehicles have a limited *driving range*, which can be fully replenished at a charging station;
- Each second-level route originates at a satellite, visits a sequence of customers and possibly some *recharging stations*, and returns to the same satellite;
- Each satellite also hosts a charging station at its location;
- Charging stations can be used multiple times, but a consecutive visit to two charging stations in a second-level route is prohibited.

The cost of the solution, to be minimized, includes a fixed cost for each vehicle in use, as well as driving costs proportional to the distance traveled.

To solve this problem, we introduce a LNS-based metaheuristic which combines a restricted set of destruction and reconstruction operators, a local search procedure, and a fast labeling algorithm to optimize the visits to charging locations. We also propose an exact mathematical programming algorithm, which uses a decomposition technique to enumerate promising first-level solutions along with bounding functions and route enumeration for the second level, using problem-tailored pricing algorithms. These two methods can be viewed as extensions of the approaches of Breunig et al. (2016) and Baldacci et al. (2013) for the classical 2EVRP, in which specialized route evaluation techniques, labeling algorithms and dominance strategies have been integrated to efficiently manage the selection of recharging stations for the electric vehicles.

Not only do these algorithms allow to find optimal or near-optimal solutions for the E2EVRP, and thus respond to the need of advanced algorithms for future city-logistics planning, but they also open the way to an analysis of several defining features of optimized battery-powered city-distribution networks. To that end, we created new datasets which simulate the general characteristics of a metropolitan area, and examine the impact of the density of the charging station network and the capacity of the vehicles' batteries on the cost-efficiency of the optimal solutions of the problem.

The remainder of this paper is organized as follows. Section 3.2 reviews the related literature and Section 3.3 formally describes the problem. Then, Sections 3.4 and 3.5 describe, respectively, the proposed exact and heuristic algorithms. Section 3.6 reports our computational experiments and sensitivity analyses, and Section 3.7 concludes.

3.2 Literature Review

We review the existing solution algorithms for the 2EVRPs, discuss the recent studies dedicated to routing optimization for vehicles with alternative fuels, and finally examine the use of en-route recharging in recent studies and applications.

Two-echelon vehicle routing problems. Several early studies focused on mathematical programming solution techniques for the 2EVRP. Gonzalez-Feliu et al. (2008) were the first to describe a branch-and-cut algorithm based on a commodity flow formulation that solved instances with up to 32 customers and 2 satellites. The method of Gonzalez-Feliu et al. (2008) was improved by Perboli et al. (2010) and Perboli et al. (2011) by adding valid inequalities in a cutting plane fashion. Optimal solutions for instances with up to 32 customers and 2 satellites were found by the method of Perboli et al. (2011). Jepsen et al. (2012) described a branch-and-cut algorithm based on a new mathematical formulation and different valid inequalities. Exact algorithms were also designed by Baldacci et al. (2013) and by Santos et al. (2015). Santos et al. described a branch-and-cut-and-price algorithm for the 2EVRP that relies on a reformulation based on the q-routes relaxation proposed for the CVRP by Christofides et al. (1981). Baldacci et al. proposed an exact method for solving the 2EVRP based on a set partitioning formulation with side constraints. They described a bounding procedure that is used by the exact algorithm to decompose the problem into a limited set of multi-depot capacitated vehicle routing problems (MDCVRPs) with side constraints. The optimal 2EVRP solution is obtained by solving the set of MDCVRPs generated. The method was tested on 207 instances, taken both from the literature and newly generated, with up to 100 customers and 6 satellites. The results obtained by Baldacci et al. (2013) show that their exact algorithm outperforms the existing methods from the papers described above. Finally, Perboli et al. (2018b) recently found new valid inequalities for the 2EVRP. Using these inequalities within a branch-and-cut algorithm allowed to solve several new instances with up to 50 customers.

The number of heuristics, metaheuristics and case studies focused on multi-echelon vehicle routing problems has also rapidly grown in the last decade. The surveys by Cuda et al. (2015) and Schiffer et al. (2019) capture well the breadth of this line of research. The former covers different two-echelon structured transportation problems: 2ELRPs, 2EVRPs and truck and trailer routing problems (TTRPs). The latter focuses on vehicle routing problems and location routing problems with intermediate stops, dedicated to replenishment, refueling or idling. Among the most recent contributions in this domain, Zeng et al. (2014) proposed a greedy randomized adaptive search procedure with a route-first cluster-second splitting algorithm and a variable neighbourhood descent for the 2EVRP. Their results are promising, but the algorithm was only tested on the smaller benchmark instances with up to 50 delivery points. Breunig et al. (2016) introduced a LNS for 2EVRPs and the 2ELRPSD. The method uses six destroy and one repair operator as well as some well-known local search procedures. It finds or improves 95% of the best known solutions for the classical benchmark instances. Given the efficiency and the effectiveness of this method, the same general structure has been used for the heuristic proposed in this paper, in addition with multiple improvements and adaptations to account for the specificities of electric vehicles. Later on, Wang et al. (2017a) studied an extension of the 2EVRP with stochastic demands, described as a stochastic program with recourse. A genetic algorithm was proposed, and the results on the problem with stochastic demands were compared to the best known deterministic solutions.

Electric vehicle routing, en-route recharging and battery swaps. Over the last decade, research has also rapidly progressed on VRPs with alternative propulsion modes: electric or hybrid. As generally reflected in the surveys of Montoya (2016); Pelletier et al. (2016) and Schiffer et al. (2019), many of these studies consider possible en-route recharging or battery swaps to overcome the range limitations of electric vehicles.

Conrad and Figliozzi (2011) were amongst the first to consider optimization techniques for electric vehicles and possible recharging stops. In the proposed *recharging* VRP, batteries can be charged at customer locations subject to additional costs. Other studies were focused on VRPs considering different aspects of environment-friendly transport. In particular, Erdogan and Miller-Hooks (2012) proposed the green vehicle routing problem (GVRP), involving battery-powered vehicles with possible en-route recharging at dedicated stations, and evaluated the implications of refueling-stations availability and dispersion.

After these seminal works, the literature progressed towards more intricate problem variants and solution methods. Schneider et al. (2014) introduced additional time-window constraints for customer deliveries as well as recharging delays. New benchmark instances were introduced, and solved by means of a hybrid heuristic combining VNS and tabu search (TS). Desaulniers et al. (2016) developed an exact method based on branch-price-and-cut, and presented computational results for the same benchmark instances.

Moreover, to progress towards real applications and improve the accuracy of the studies, other important characteristics of real delivery networks have been considered. Heterogeneous fleets with different propulsion modes were studied in Felipe et al. (2014), jointly with a heterogeneous set of recharging stations with different cost and recharging time. Goeke and Schneider (2015) considered a mix of conventional and electric vehicles, evaluating the energy consumption of an electric vehicle as a function of speed, gradient and cargo load distribution. Hiermann et al. (2016) proposed the electric fleet size and mix problem with fixed costs and time windows (EFSMFTW), in which the deliveries can be performed with a mix of vehicle types, differing in their acquisition cost, freight capacity, and battery size. Experiments were conducted with a branch-and-price algorithm and a hybrid heuristic. This research was extended in Hiermann et al. (2019) to study the impact of additional plug-in hybrid vehicles.

Keskin and Çatay (2016) introduced an ALNS for a problem variant in which partial recharging is allowed.

In a recent case study, Wang et al. (2017c) stressed several interesting facts regarding the use of large battery powered commercial vehicles. They show that in the real-world transit network based in Davis, California, range anxiety can be mitigated by adopting good recharging strategies, and that several companies adapt much stricter range limits than the theoretical ones (typically half of the range) to extend battery life cycles. Finally, they indicate a list of case studies in which battery swapping has been applied to electric bus transit systems, a strategy which is justified by the fact that fast charging techniques can significantly extend vehicle ranges within only five to ten minutes of charging. Other recent works have proposed to optimize fast or partial charging strategies. In particular, Schiffer and Walther (2017a) introduced the electric location routing problem with time windows and possible partial recharging stops, whereas Keskin and Çatay (2018) extended the electric vehicle routing problem with time windows (EVRPTW) with different types of stations (super-fast, fast, and normal ones) for en-route recharging.

Real-world applications of en-route recharging. As early as 2008, electric buses were installed for the Beijing Olympic Games. Each bus, of a capacity of 50 seats, can undergo battery swapping up to three times a day (EV World, 2008). Similarly, fully electric public transit buses have been operating in Vienna, Austria for several years now. These buses stop for approximately 15 minutes at the end of their route to traverse the inner city and recharge the batteries with an overhead system multiple times a day (Wr. Linien, 2013). Schiffer et al. (2016) also recently established a case study of "TEDi Logistik GmbH & Co. KG", which provides freight transportation services and relies on two electric vehicles. They concluded that the operations would be significantly improved if fast-charging stations were additionally available at a few customers locations for en-route recharging, therefore allowing to reach locations located further than 70km from the depot. Finally, `JD.com`, one of China's biggest e-commerce companies, is replacing its existing fleet with fully electric vehicles (John, 2017). Seeing the need for it, they recently launched a contest for optimization algorithms capable of routing vehicles with en-route recharging (JD.com, 2018).

Our study shares the same objectives as many aforementioned papers: bridging the gap between academic electric VRPs and real problem attributes. The current literature on electric vehicles has only considered simplistic delivery networks with a single depot and a single echelon, but it is well known that city logistics usually involve richer configurations, with interconnected echelons and transportation modes. Moreover, the restricted range of the electric vehicles and their possible need for en-route recharging bring new challenges which have to be considered when selecting intermediate facilities (i.e., satellite) locations. We therefore propose to study the impact of electrical fleets in second-level routes, in a two-echelon delivery setting, where electric vehicles are likely to be needed.

3.3 Problem Description

The E2EVRP addressed in this paper can be formally described as follows.

A mixed graph $G = (N, E, A)$ is given, where the vertex set N is partitioned as $N = \{0\} \cup N_S \cup N_C \cup N_R$. Vertex 0 represents the depot, $N_S = \{1, 2, \dots, n_s\}$ represents n_s satellites,

$N_C = \{n_s + 1, \dots, n_s + n_c\}$ represents n_c customers, and $N_R = \{n_s + n_c + 1, \dots, n_s + n_c + n_r\}$ represents n_r charging stations. The edge set E is defined as $E = \{\{0, j\} : j \in N_S\} \cup \{\{i, j\} : i, j \in N_S, i < j\}$ and the arc set A as $A = \{(i, j) : i, j \in N_S \cup N_C \cup N_R, i \neq j\} \setminus \{(i, j) : i, j \in N_S\} \setminus \{(i, j) : i, j \in N_R\}$. A travel or routing cost d_{ij} is associated with each edge $\{i, j\} \in E$ and with each arc $(i, j) \in A$.

Each customer $i \in N_C$ requires a supply of q_i units of goods to be delivered from the depot using the following two types of vehicles. A fleet of m^1 vehicles of capacity Q_1 located at depot 0 and a fleet of m_k vehicles of capacity $Q_2 < Q_1$ located at satellite $k \in N_S$. Moreover, at most $m^2 \leq \sum_{k \in N_S} m_k$ second-level vehicles can be used.

A 1st-level vehicle route is a simple cycle in G passing through the depot and a subset of satellites such that the total demand delivered is less than or equal to Q_1. A satellite $k \in N_S$ can be visited by more than one 1st-level route and has a capacity B_k that limits the demand that can be delivered to it.

A 2nd-level route is a circuit in G passing through a satellite and a subset of customers and charging stations and such that the total demand of the visited customers does not exceed the vehicle capacity Q_2 and the following charging station constraints are respected. Each vehicle on the 2nd-level has a maximum battery capacity L, and a battery consumption c_{ij} is associated with each arc $(i, j) \in A$; the maximum battery consumption of a vehicle without a visit to a charging station is therefore equal to L. Charging stations can be visited right after or before a satellite, or in between customers, and, whenever a charging station is visited, a vehicle is fully charged up to level L. In the scope of this short-haul problem, we prohibit a consecutive visit to two charging stations.

Fixed costs U_1 and U_2 are also associated with the use of 1st-level and 2nd-level vehicles, respectively. The cost of a route (1st-level or 2nd-level) is equal to the sum of the costs of the traversed edges or arcs plus the fixed cost.

The problem asks to design both 1st-level and 2nd-level routes so that the quantity delivered from each satellite is equal to the quantity received from the depot, each customer is visited exactly once, and the total cost of the routes is minimized.

Multigraph reformulation. The E2EVRP can be reformulated as a routing problem on a multigraph $G' = (N', E', A')$, where $N' = \{0\} \cup N_S \cup N_C$ is the vertex set, $E' = \{\{0, j\} : j \in N_S\} \cup \{\{i, j\} : i, j \in N_S, i < j\}$ is the edge set and A' is the arc set. Arc set A' is used to represent 2nd-level routes and is defined as $A' = \{(i, j) : i, j \in N_S \cup N_C, i \neq j\} \setminus \{(i, j) : i, j \in N_S\}$. The arc set A' also contains the following set of arcs:

- With each arc $(i, j) \in A'$ are associated $h(i, j)$ arcs representing the different paths that a 2nd-level vehicle can take to go from vertex i to vertex j with at most one charging station visited in between vertices i and j.

- A *cost* $d(i, j, p)$, a *consumption* $c(i, j, p)$ and a *charging station* $s(i, j, p) \in N_R$ are associated with each arc (i, j, p), $p = 1, \dots, h(i, j)$, $\forall (i, j) \in A'$. We assume that $s(i, j, p) = 0$ if arc (i, j, p) represents the direct path (i, j) without any charging station visited in between i and j.

- The cost $d(i,j,p)$ and the consumption $c(i,j,p)$ of arc (i,j,p) are defined as follows:

$$\begin{cases} d(i,j,p) = d_{ij},\ c(i,j,p) = c_{ij}, & \text{if } s(i,j,p) = 0 \\ d(i,j,p) = d_{ik} + d_{kj},\ c(i,j,p) = c_{kj}, & \text{if } k = s(i,j,p) \neq 0. \end{cases}$$

Multigraph G' does not contain any arc (i,j,p) such that $s(i,j,p) = 0$ and $c_{ij} > L$, or $k = s(i,j,p) \neq 0$ and $c_{ik} > L$ or $c_{kj} > L$. Notice that for arc (i,j,p) with $s(i,j,p) \neq 0$ value $L - c(i,j,p)$ represents the battery level of the vehicle after arriving at vertex j whereas if $s(i,j,p) = 0$, i.e., no charging station is visited in between i and j, the battery level at vertex j is equal to $b - c(i,j,p)$, where b is the battery level at vertex i.

A 2nd-level route for satellite $k \in N_S$ in graph G' is a simple circuit in G' passing through a satellite and a subset of customers and such that (i) the total demand of the visited customers does not exceed the vehicle capacity Q_2 and (ii) the vehicle leaves satellite k with a consumption equal to 0 (or, equivalently, the vehicle is fully charged) and its consumption at each visited vertex does not exceed the maximum battery capacity L.

The following proposition holds.

Proposition 1 *There is a one-to-one correspondence between 2nd-level routes in G and 2nd-level routes in G'.*

Moreover, the set of arcs A' can be reduced by means of the following dominance rule.

Proposition 2 *An optimal E2EVRP solution cannot contain an arc (i,j,r_1) if there exists another arc (i,j,r_2), $r_1 \neq r_2$, such that:*

1. *$i \in N_S$: $d(i,j,r_1) \geq d(i,j,r_2)$ and $c(i,j,r_1) \geq c(i,j,r_2)$;*
2. *$i \in N_C$: $d(i,j,r_1) \geq d(i,j,r_2)$ and $c(i,j,r_1) \geq c(i,j,r_2)$, and $c_{ik_1} \geq c_{ik_2}$, $k_1 = s(i,j,r_1)$, $k_1 \neq 0$, and $k_2 = s(i,j,r_2)$, $k_2 \neq 0$.*

3.4 Solving the E2EVRP to Optimality

The method used to solve the E2EVRP to optimality is based on the exact method proposed by Baldacci et al. (2013) for the 2EVRP. More precisely, we tailored the method described by Baldacci et al. to handle the multigraph G' described in Section 3.3. The exact method consists of the following two main steps.

1. The set of all 1st-level routes is generated and a lower bound LB0 on the E2EVRP is computed. The computation of LB0 is based on a integer relaxation that results in a multiple-choice knapsack problem. In computing lower bound LB0, we extended the *ng*-routes relaxation used in Baldacci et al. to the case of our multigraph G' (see below).

2. The set of all possible subsets of 1st-level routes that could be used in any optimal E2EVRP solution is generated. For each subset of 1st-level routes the following steps are executed:

   (i) Lower bound LB0 is computed by fixing the selected set of 1st-level routes in the solution. If the resulting lower bound is greater than or equal to the cost of the best incumbent E2EVRP solution, then the current subset is rejected, otherwise the next step is executed;

(ii) The E2EVRP problem obtained by considering only the selected set of 1$^{\text{st}}$-level routes is solved to optimality. The resulting problem is a MDCVRP, that is solved using the method proposed by Baldacci and Mingozzi (2009). The optimal solution cost of the E2EVRP corresponds to the minimum solution cost of such MDCVRPs. In solving problem MDCVRP, we extended the procedure used to generate elementary routes described in Baldacci and Mingozzi (2009) to the case of our multigraph G'.

The procedure is initialized with the best upper bound computed by the heuristic algorithm described in Section 3.5. In the computational results of Section 3.6, we will denote with LB1 (LB2) the minimum of the lower bounds computed at Step 2-(i) (Step 2-(ii)) over the set of subsets of 1$^{\text{st}}$-level routes. Lower bound LB2 is computed using the lower bounds provided by the method of Baldacci and Mingozzi (2009).

In the following, we describe how we extended the *ng*-routes relaxation to graph G' and, for the sake of space, we omit the details of the procedure used to generate elementary routes, which is indeed a straightforward adaptation of the procedure used by Baldacci and Mingozzi (2009).

Pricing ng-routes. The computation of the lower bounds at steps 1, 2-(i) and 2-(ii) and the procedure used to generate elementary routes rely on the use of the *ng*-routes relaxation. In this section, we describe the extension of the relaxation described in Baldacci et al. (2011) to multigraph G'. We describe the relaxation for a generic satellite $k \in N_S$ that, for sake of notation, is denoted with the index 0 in the description reported below.

Let $\Omega(w, j, i, p)$ be the subset of battery consumption values from vertex j to arrive at vertex i with a consumption equal to w, with $w \leq L$, when j is visited immediately before i using arc of index p of arc $(j, i) \in A'$. Set $\Omega(w, j, i, p)$ is defined as follows:

$$\Omega(w, j, i, p) = \begin{cases} \{w - c_{ji}\} & \text{if } s(j, i, p) = 0 \text{ and } c_{ji} \leq w \\ \{w' : 0 \leq w' + c_{jk} \leq L\} & \text{if } s(j, i, p) = k \neq 0 \text{ and } c_{ki} = w \\ \varnothing & \text{otherwise.} \end{cases} \tag{10}$$

Let $N_i \subseteq N_C$ be a set of selected customers for vertex i such that $N_i \ni i$ and $|N_i| \leq \Delta(N_i)$ ($\Delta(N_i)$ is an a priori defined parameter). The sets N_i allow us to associate with each forward path $P = (0, i_1, \dots, i_k)$ in G' the subset $\Pi(P) \subseteq V(P)$, $V(P) = \{0, i_1, \dots, i_{k-1}, i_k\}$, containing customer i_k and every customer i_r, $r = 1, .., k-1$, of P that belongs to all sets $N_{i_{r+1}}, \dots, N_{i_k}$ associated with the customers $i_{r+1}, \dots, i_k$ visited after i_r. The set $\Pi(P)$ is defined as: $\Pi(P) = \{i_r : i_r \in \bigcap_{s=r+1}^{k} N_{i_s}, r = 1, \dots, k-1\} \cup \{i_k\}$.

An *ng-path* (NG, q, w, i) is a non-necessarily elementary path $P = (0, i_1, \dots, i_{k-1}, i_k = i)$ starting from the satellite 0 with an initial consumption equal to 0, visiting a subset of customers (even more than once) of total demand equal to q such that $NG = \Pi(P)$, ending at customer i with a total consumption equal to w, and such that $i \notin \Pi(P')$, where $P' = (0, i_1, \dots, i_{k-1})$ is an *ng*-path. We denote by $f(NG, q, w, i)$ the cost of the least cost *ng*-path (NG, q, w, i). An (NG, q, w, i)-route is an $(NG, q, w, 0)$-path where i is the last customer visited before arriving at the satellite.

Functions $f(NG, q, w, i)$ can be computed using dynamic programming (DP). The state space graph $\mathscr{H} = (\mathscr{E}, \Psi)$ is defined as follows: $\mathscr{E} = \{(NG, q, w, i) : q_i \leq$

$q \leq Q_2, \forall NG \subseteq N_i$ s.t. $NG \ni i, \sum_{j \in NG} q_j \leq Q_2, \forall i \in \{0\} \cup N_C, \forall w, 0 \leq w \leq L\}$, $\Psi = \{((NG', q', w', j), (NG, q, w, i))^p : \forall (NG', q', w', j) \in \Psi^{-1}(NG, q, w, j, i, p), p = 1, \ldots, h(j,i), \forall (j,i) \in A', \forall (NG, q, w, i) \in \mathscr{E}\}$, where $\Psi^{-1}(NG, q, w, j, i, p) = \{(NG', q - q_i, w', j) : \forall NG' \subseteq N_j$ s.t. $NG' \ni j$ and $NG' \cap N_i = NG \setminus \{i\}$, $\forall w' \in \Omega(w, j, i, p)\}$.

The DP recursion for computing $f(NG, q, w, i)$ is:

$$f(NG, q, w, i) = \min_{\substack{(j,i) \in A',\ 1 \leq p \leq h(j,i) \\ (NG', q', w', j) \in \Psi^{-1}(NG, q, w, j, i, p)}} \{f(NG', q', w', j) + d(j, i, p)\}, \forall (NG, q, w, i) \in \mathscr{E}, \quad (11)$$

using as initial state $f(\{0\}, 0, 0, 0) = 0$ and $f(\{0\}, q, w, 0) = \infty$ for $q > 0$ and $w > 0$. In the computational experiments (Section 3.6), we set $\Delta(N_i) = 12$, $\forall i \in N_C$, and N_i contains i and the 11 nearest customers to i.

From the experimental analysis presented in Section 3.6, we observed that this mathematical programming algorithm can solve small and medium size instances to optimality and provide good lower bounds otherwise. Yet, this method requires a good initial upper bound to be truly effective, especially when the problem size grows. To produce these upper bounds, the following section introduces a metaheuristic based on large neighborhood search.

3.5 Large Neighborhood Search

Our metaheuristic, called LNS-E2E, follows the basic principles of ruin and recreate (Shaw, 1998). At each iteration, some parts of the solution are destroyed by a selected destroy operator (Section 3.5.1), and then repaired again (Section 3.5.2) with a three-steps repair operator which reconstructs, in turn, the 2$^{\text{nd}}$-level routes, the 1$^{\text{st}}$-level routes, and completes the reconstruction with an optimal insertion of visits to charging stations. Subsequently, a sophisticated local search (Section 3.5.3) is applied to improve the resulting solution. During the local search, the labeling algorithm is used in combination with the moves to evaluate their impact.

The general structure of the method is presented in Algorithm 2. The sequence of destruction, reconstructions and local search is repeated until i_{max} iterations have been performed without improvement of the incumbent solution (Lines 4–8). Once this termination criterion is attained, the best solution is stored (Lines 9–10) and the method performs a restart from a new random initial solution. This process repeats until a maximum time T_{MAX} is attained (Lines 2–10).

In contrast with the adaptive large neighborhood search of Pisinger and Ropke (2007), LNS-E2E makes uses of a very limited number of destroy operators, and a single repair operator. Moreover, the probabilities of use of each operator are fixed, i.e., the method does not rely on adaptive mechanisms. This design is in line with the study of Breunig et al. (2016), where it was observed that the algorithm with a simple fixed probability selection equaled its adaptive counterpart on the 2EVRP. The following subsections now describe each component of the method in deeper details.

3.5.1 Destroy operators

At each iteration, one out of three destroy operators is selected with equal probability:

Algorithm 2: LNS-E2E

```
$\mathcal{S}^{best} \leftarrow \varnothing$
while CPU time $< T_{\text{MAX}}$ do
    $\mathcal{S} \leftarrow$ LocalSearch(Repair($\varnothing$))          /* (re-)start: new solution */
    for $i \leftarrow 0$ to $i_{max}$ do
        $\mathcal{S}^{temp} \leftarrow$ LocalSearch(Repair(Destroy($\mathcal{S}$)))
        if Cost($\mathcal{S}^{temp}$) $<$ Cost($\mathcal{S}$) then
            $\mathcal{S} \leftarrow \mathcal{S}^{temp}$          /* accept better solution */
            $i \leftarrow 0$
    if Cost($\mathcal{S}$) $<$ Cost($\mathcal{S}^{best}$) then
        $\mathcal{S}^{best} \leftarrow \mathcal{S}$          /* store best solution */
return $\mathcal{S}^{best}$
```

- **A) Related nodes removal.** A seed customer is randomly chosen. A random number of its Euclidean closest customers as well as the seed customer are removed from the current solution and added to the list of nodes to re-insert. This operator receives a parameter p_1, which denotes the maximum percentage of nodes to remove. At most $\lceil p_1 \cdot n_c \rceil$ nodes are removed.
- **B) Random routes removal.** Randomly selects routes and removes the associated customers visited, adding them to the list of nodes to re-insert. This operator randomly selects a number of routes in the interval $[0, \lceil p_2 \cdot \frac{q_{\text{TOT}}}{Q_2} \rceil]$. The last term gives a lower bound on the number of routes needed to serve all customers.
- **C) Close satellite.** Chooses a random satellite. If the satellite can be closed and the remaining open ones still can provide sufficient capacity for a feasible solution, the chosen satellite is closed temporarily. All customers that are assigned to it are removed and added to the list of nodes to re-insert. The satellite stays closed until it is opened again in a later phase.

Moreover, the following two other operators may be applied right after one of the destroy operators described above:

- **D) Open all satellites.** With a probability of $\hat{p}_3$, all currently closed satellites are set to be available again in future repair phases.
- **E) Remove single customer routes.** This operator removes all routes which contain only one single customer. Typically, a complete solution does not often contain any route matching this criterion, but this can happen after a partial destruction. Therefore, with a probability of $\hat{p}_4$, all those customers which remain on a single node route after the destruction phase are also added to the list to re-insert. As there is a limit on the number of vehicles available, removing short routes also allows to use a vehicle originating from another satellite in the next repair phase.

3.5.2 Repair operator and initial solution construction

We propose a repair operator based on three steps, which first reinserts customer-visits in 2$^{\text{nd}}$-level routes, then reconstructs 1$^{\text{st}}$-level routes, and finally completes the solution with recharging stations visits. Note also that the creation of the initial solution can be seen as a totally destroyed or empty solution ($\varnothing$), and therefore follows the same principle.

Reinsertion of customer visits. The classic cheapest insertion calculates every possible insertion position for every node to insert and selects the least-cost one. LNS-E2E uses a simplified version of this greedy heuristic with lower complexity, which iteratively inserts the first node from the insertion list in its cheapest position, until all nodes have been inserted. As a consequence, the outcome of the reconstruction depends on the order of the nodes in the list, favouring diversification.

The order of nodes in the list is randomly shuffled prior to insertions. In the exceptional case where this method fails to generate a feasible solution, another construction is attempted, this time ordering the nodes in the list by decreasing demand quantity. Such an ordering has a better chance to result in a feasible solution with respect to the capacity (i.e., packing) constraints, since no split deliveries are allowed on the second level. Overall, this first phase of the 2$^{\text{nd}}$-level routes reconstruction respects all constraints of the problem except those related to charging levels and recharging station visits.

Construction of first level tours. After itineraries for the 2$^{\text{nd}}$-level routes have been found, the quantities needed at the satellites are known. With this information, the 1$^{\text{st}}$-level routes can be reconstructed. We opted for a complete reconstruction, as the number of satellites is usually small and the 2$^{\text{nd}}$-level routes can very significantly change from one iteration to another. On the first level, split deliveries are not only allowed, but sometimes also necessary to find a feasible solution. Depending on the customers associated to a satellite, it can occur that the requested quantity at the satellite is larger than a full truckload. Therefore, we use a simple preprocessing step: for any satellite with a demand larger than a full truckload, we create a back-and-forth trip from the depot, until the remaining demand is smaller than a truck's capacity. The simplified cheapest insertion is then used to complete the 1$^{\text{st}}$-level solution. In practical settings with up to 10 or 20 satellite facilities, this method finds optimal 1$^{\text{st}}$-level routes in a majority of cases in a very limited computational effort.

Optimal insertion of charging stations visits. At this point, the algorithm has reconstructed a solution which is feasible in terms of load capacities but usually infeasible in terms of battery capacities. To restore feasibility, it uses a DP algorithm which finds the optimal charging stations positions for each 2$^{\text{nd}}$-level route. The problem of inserting charging station visits in a route $\boldsymbol{\sigma} = (\sigma_0, \sigma_1, \dots, \sigma_K)$ can be reduced to a shortest path problem with resource constraints (SPPRC) in a directed acyclic multigraph $\bar{H} = (\bar{N}, \bar{A})$, such that $\bar{N} = \{0\} \cup \{1, \dots, K-1\} \cup \{K\}$. The nodes 0 and K correspond to depot visits (such that $\sigma_0 = \sigma_K = 0$), while the other nodes represent customer visits. Each arc $(i, i+1, r_k) \in \bar{A}$ corresponds to a non-dominated arc between σ_i and σ_{i+1}, with the same characteristics as $(\sigma_i, \sigma_{i+1}, r_k) \in A'$ defined in Section 3.3 and possible en-route recharging. This multigraph is illustrated in Figure 4. You can find a numerical example in the Appendix in Section 3.8.1.

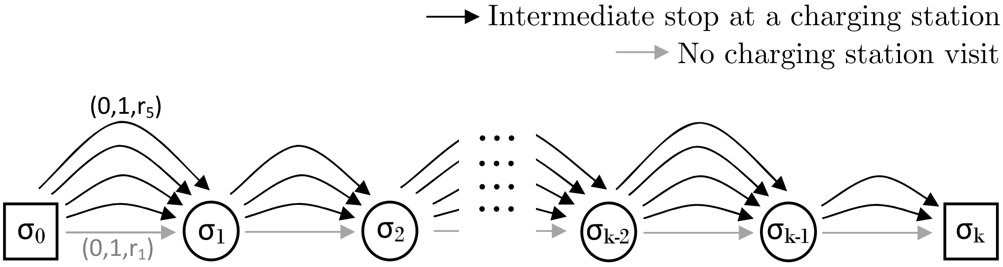

Intermediate stop at a charging station
No charging station visit
(0,1,r5)
(0,1,r1)
σ0
σ1
σ2
σk-2
σk-1
σk

Figure 4: Illustration of the multigraph $\bar{H}$. Non-dominated choices of charging stations visits are represented by parallel arcs.

Solving this SPPRC can be simply done using Bellman's algorithm in the topological order $(0, 1, \ldots, K)$. Using the same notations as in previous sections, the state space graph $\bar{\mathscr{H}} = (\bar{\mathscr{E}}, \bar{\Psi})$ is defined as $\bar{\mathscr{E}} = \{(w, i) : \forall w, 0 \leq w \leq L, \forall i \in \bar{N}\}$ and $\bar{\Psi} = \{((w', i-1), (w, i))^p : \forall (w', i-1), w' \in \Omega(w, \sigma_{i-1}, \sigma_i, p), p = 1, \ldots, h(\sigma_{i-1}, \sigma_i), \forall (w, i) \in \bar{\mathscr{E}}\}$. Defining $f(w, i)$ as the minimum cost of a path starting from 0 and reaching node i with battery consumption w, the DP recursion can be expressed as:

$$f(w, i) = \min_{\substack{1 \leq p \leq h(\sigma_{i-1}, \sigma_i) \\ w' \in \Omega(w, \sigma_{i-1}, \sigma_i, p)}} \{f(w', i-1) + d(\sigma_{i-1}, \sigma_i, p)\}, \forall (w, i) \in \bar{\mathscr{E}}, \tag{12}$$

and the initial state is set as $f(0, 0) = 0$ and $f(w, 0) = \infty$ for $w > 0$.

In the rare case where no feasible path can be found at the end of the DP recursion, a second execution of the DP algorithm is done, with a minor modification of the label propagation function allowing to use and penalize battery capacity excesses. In this case, any consumption over the battery level $w > L$ is converted into a proportional penalty of $M \times (w - L)$, where M is a large constant. Therefore, the method seeks a route with the smallest penalty in priority, and then the shortest distance. Due to its large impact on the objective, this infeasibility will generally be resolved in the next steps of the method: the local search or the next destroy and repair phase.

3.5.3 Local search with systematic charging stations relocations

After solution reconstruction, LNS-E2E applies a local search procedure on the 2nd-level routes based on 2-OPT, 2-OPT*, RELOCATE, SWAP and SWAP2-1 moves (see, e.g., Vidal et al., 2013, for a detailed description of these neighbourhoods). The 2-OPT* moves are only tested between routes originated from the same satellite. Moreover, when testing moves that involve routes from different satellites, the algorithm checks that enough capacity is available in the satellites and the associated 1st-level routes. The moves are tested in random order and a first-improvement acceptance policy is used, i.e., any move which results in an improvement in terms of cost is directly applied, until no more improvement can be found. Similarly to the granular search by Toth and Vigo (2003), the moves are limited to node pairs (i, j) such that j belongs to one of the Γ closest vertices from i.

Most modifications of the sequence of customer visits or their assignment to vehicles induce some necessary changes in the planning of charging stations visits. Ideally, one would like to apply the labeling algorithm described in Section 3.5.2 to obtain the *exact* cost of each move, with an optimal placement of recharging stations in each newly-created route. Such an evaluation would be, however, prohibitive in terms of computational effort. To speed up the method with only a minimal impact on solution quality, we propose some heuristic move filters, which are quite similar in principle to the techniques used by Taillard et al. (1997) for the VRP with soft time windows. We first evaluate each move without the labeling algorithm to obtain an approximation of its impact on the total distance. When doing this calculation, the current locations of the charging stations are unchanged. Any move which is feasible in terms of load capacity and does not deteriorate the total distance by more than 3% is then evaluated exactly in combination with the labeling algorithm, so as to find better charging stations locations which may lead to an improvement. After this exact evaluation, any improving move is applied.

3.6 Computational Experiments

We conducted extensive experimental analyses with two aims. Firstly, we evaluate the performance of the proposed algorithms for different types of instances, and measure the benefit of integrated routing and recharging-stations planning (Section 3.6.1). For this analysis, we extend classical 2EVRP instances into E2EVRP instances in order to obtain diverse and challenging datasets and allow possible comparisons with previous algorithms. Secondly, we analyze the impact of two defining features of electric-fueled city-distribution networks: the density of charging stations in a city, and the vehicles' battery capacities (Section 3.6.2). For this analysis, we produced a new set of medium-scale instances which simulates a realistic delivery scenario in a metropolitan area, using battery specifications from recent electric vehicles.

The mathematical programming algorithm was coded in Fortran 77, and run on a single thread of a 3.6 GHz Intel i7-4790 CPU with 32GB of RAM. It relies on CPLEX 12.5.1 for the resolution of the linear programs and some integer subproblems. The metaheuristic was coded in Java (JRE 1.8.0–151), and run on a single thread of a 3.4 GHz Intel i7-3770 CPU. All benchmark instances used in this paper are available for download at `https://www.univie.ac.at/prolog/research/electric2EVRP` and `https://w1.cirrelt.ca/~vidalt/en/VRP-resources.html`.

3.6.1 Method performance and benefits of integrated planning

Our benchmark instances for this first analysis are natural extensions of the 2EVRP instances known as Set 2 and 3 by Perboli et al. (2011), Set 5 by Hemmelmayr et al. (2012), and Set 6 by Baldacci et al. (2013). The depot, satellite and customers locations remain unchanged. The new information is associated to the electric vehicles (maximum charging level and energy consumption) and to the charging stations (coordinates).

The selection of charging stations follows the guidelines of Schneider et al. (2014) (instances of the electric vehicle routing problem with time windows and recharging stations (EVRPTWRS)) and Desaulniers et al. (2016). The ratio between charging stations and customers is chosen between 1/10 and 1/5. Firstly, every depot and satellite location provides charging abilities to vehicles. To pick the remaining locations, we defined a grid of

100×100 candidate locations based upon the range of x- and y-axis coordinates from the existing 2EVRP instances. For each location, we counted the number of customers in "close proximity", defined as half the average tour length in the best known 2EVRP solution. The more customers one candidate location has in proximity, the more likely it is to be selected as a charging station. This was achieved by a roulette wheel selection of the remaining charging stations among those 10,000 locations. Finally, all distances are calculated as Euclidean and rounded to the nearest integer value. To reduce the effect of rounding, all x- and y-coordinates from the classical 2EVRP instances have been multiplied by a factor ten. For each instance, the battery capacity has been defined as $L = \max\{0.6\,\gamma_1, 2.0\,\gamma_2\}$, where γ_1 represents the average route length of all second-level routes in the best-known 2EVRP solution, and γ_2 is the largest distance of a customer to its closest recharging station. For the sake of simplicity, the energy consumption per distance unit is always set to 1 ($d_{ij} = c_{ij}$). As highlighted in our computational experiments, this approach leads to feasible solutions for all the instances, whereas the best-known 2EVRP solutions are generally not feasible for the corresponding E2EVRP instances.

Parameters calibration. To produce suitable values for the new parameters of the LNS, we used a preliminary meta-calibration based on the CMA-ES of Hansen (2006). During meta-calibration, the parameters are considered to be the decision variables, and the associated objective corresponds to the average solution quality of LNS-E2E over ten runs on a set of training instances. This training set includes six larger-scale instances from Set 5: {100-5-1, 100-5-2b, 100-10-1, 100-10-2b, 200-10-1, 200-10-2b}. Table 15 lists the method parameters, the allowed range for each parameter, and the final values found by the meta-calibration process. These values will be used for the rest of the experiments.

After calibration, we evaluated the proposed mathematical programming algorithm and the metaheuristic on the complete set of benchmark instances. The termination criterion of LNS-E2E has been set to $T_{\text{MAX}} = 150$ seconds for the small instances of Sets 2, 3 and 6a, and 900 seconds for the large-scale instances of Set 5.

Table 15: Range of parameters used during meta-calibration, and final values found

	Parameter	Search interval	Final value
p_1	Related removal (%)	0–100	11
p_2	Random route removal (%)	0–100	37
$\hat{p}_3$	Open all satellites (%)	0–100	12
$\hat{p}_4$	Remove single customer routes (%)	0–100	18
τ	Granularity threshold for move evaluations	0–40	25
i_{max}	Number of non-improving iterations before restart	0–1000	385

Results on small instances. Table 16 reports the results on the smaller instances of Set 2 and Set 3 with 21 customers. The leftmost group of columns reports the characteristics of the instances: number of customers n_c, satellites n_s, trucks m^1, (electric) second level vehicles m^2 and charging stations n_r. The next group of columns shows the performance of the mathematical programming algorithm. The column UB reports the best upper bound at the end of the algorithm, and the solutions marked with an asterisk are proven optimal. The lower bounds obtained at different steps of the exact method are also displayed along with the associated CPU time values, using the same notations as described in Section 3.4. The next group of columns reports the performance of LNS-E2E: the average (Avg) and best (Best) solution quality over five runs, the average computational time per run (T(s)), and the average time per run needed to reach the final solution of the run (T*(s)). The overall best known solution (BKS) found during all experiments (including calibration and testing) is reported in the rightmost column.

The exact algorithm produced optimal solutions (marked with an asterisk) for all instances except one. A notable improvement is visible when comparing LB2, obtained by repeated resolutions of MDCVRP with side constraints, with LB0 and LB1. The CPU time of the method remains below one minute for 4/12 instances, but can rise up to six hours in other cases, illustrating the difficulty of the E2EVRP, as the presence of the battery capacity constraints significantly increases the time needed for route enumeration. For these instances, the metaheuristic always found the optimal solutions in at least one run out of five. Still, we observe some variance in the results of different independent runs. This is due to the combination of multiple classes of decision variables (two-echelon routing, satellite selection and charging stations selection), which make the problem very intricate and favors the creation of many local minima. The LNS-E2E remains nonetheless accurate, with an average gap of 1.18% from the optimal or best known solutions. The time taken to attain the final solution of the run varies from 2 to 132 seconds, depending on the instance.

Results on medium instances. Table 17 displays the results on the medium instances of Sets 2, 3 and 6a, containing between 32 to 75 customers. The same convention as the previous table is used. For this scale of instances, the mathematical programming algorithm does not generate proven optimal solutions in the allotted time and was stopped after the computation of bound LB1. The average optimality gap between the best solution found by LNS-E2E and the bound LB1, for this group of instances, amounts to 3.57%, demonstrating the good accuracy of both approaches. As usual when comparing exact algorithms with metaheuristics, the difference of CPU time between the two methods becomes more marked for larger instances. For some instances with 75 customers, the time needed to compute LB1 grows as high as 25 hours, whereas the termination of the heuristic is guaranteed after 150 seconds.

Results on large instances, and impact of integrated routing and recharging stations optimization. The larger instances of Set 5 contain 100 or 200 customer visits. To the best of our knowledge, only 3/18 instances have been solved to proven optimality for the classical 2EVRP (without considering electric vehicles and recharging stations). The E2EVRP appears to be even harder to solve, and our exact approach could not produce optimal solutions or good lower bounds in reasonable time. For this set of instances, we therefore concentrate our analysis on the results of the metaheuristic, with the aim of assessing the performance of the algorithm and the impact of an integrated optimization of routing and recharging stations decisions. To that end, we compared four algorithms. The first two algorithms solve the 2EVRP *without* electric vehicles:

- **LNS-2E**: the algorithm presented in Breunig et al. (2016) (LNS-2E), which produces the current state-of-the-art results for that problem;
- **LNS-E2E ∞**: the proposed algorithm, in which the range of the electric vehicles is set to ∞ to make recharging-stations visits unnecessary.

The two other solution methods are designed for the problem *with* electric vehicles:

- **LNS-2E post**: resolution of the classical 2EVRP (disregarding battery constraints) with LNS-2E, followed by a post-optimization using the labeling algorithm to insert charging-stations visits;
- **LNS-E2E**: the proposed algorithm, with integrated routing and planning of charging stations.

Each method was run until a time limit of 15 minutes, and the same rounding convention (integer distances) have been adopted to allow direct solution comparisons. Table 18 reports the average (Avg) and best (Best) solution quality of each method over ten runs, as well as the average CPU time to reach the final solution of each run (T*(s)). For future reference, the BKS found on Set 5 for the LNS-E2E during preliminary calibration and testing are also listed in the rightmost column.

Firstly, these results highlight the good accuracy of the proposed LNS-E2E, even for the particular case of the 2EVRP *without* electric vehicles. In comparison with the current state-of-the-art algorithm LNS-2E, better average quality solutions are found on all 200-customer instances, with improvements rising up to 2.41%, while solutions of slightly lower quality are obtained on the 100-customer test cases.

Secondly, we observe the large benefits of an integrated routing and charging stations visits planning. Even when starting with good 2EVRP solutions, a post-ex insertion of charging stations results in solutions of poor quality for the E2EVRP in comparison with the integrated LNS-E2E approach. The average gain related to an integrated optimization in comparison to post-optimization amounts to 3.28%, and can reach as high as 7.93% for instance 100-10-2b. Finally, in terms of computational effort, we observe that the proposed LNS-E2E approach finds solutions in a similar time as LNS-2E, despite the joint optimization of charging stations. This is essentially due to the heuristic move filters described in Section 3.5.3, allowing to evaluate and discard a large proportion of non-promising local search moves without a call to the labeling algorithm. The next section will study the impact of some of these method components in deeper detail.

Table 16: Performance analysis on small instances of Set 2 and 3

	Characteristics					Exact Method									LNS-E2E				
Instance	n_c	n_s	m^1	m^2	n_r	UB	%LB0	LB0	$\mathrm{T}_{\mathrm{LB0}}$(s)	%LB1	LB1	%LB2	LB2	$\mathrm{T}_{\mathrm{All}}$(s)	Avg	Best	T(s)	T*(s)	BKS
Set2																			
n22-k4-s6-17	21	2	3	4	4	5229*	98.23%	5136.3	2.3	98.87%	5169.9	100.00%	5229.00	5.6	**5229.0**	**5229**	150	6.1	**5229**
n22-k4-s8-14	21	2	3	4	4	5094*	95.95%	4887.6	2.9	95.95%	4887.6	100.00%	5094.00	77.2	5168.4	**5094**	150	39.8	**5094**
n22-k4-s9-19	21	2	3	4	4	5236*	93.95%	4919.1	4.3	93.95%	4919.1	99.62%	5216.26	14.3	5240.2	**5236**	150	65.3	**5236**
n22-k4-s10-14	21	2	3	4	4	5561*	96.99%	5393.6	4.2	96.99%	5393.6	99.64%	5541.15	343.9	**5561.0**	**5561**	150	2.0	**5561**
n22-k4-s11-12	21	2	3	4	4	5793*	95.92%	5556.6	3.6	95.92%	5556.6	99.24%	5748.74	34.1	5822.0	**5793**	150	74.8	**5793**
n22-k4-s12-16	21	2	3	4	4	4125*	96.96%	3999.5	5.6	97.17%	4008.1	100.00%	4125.00	8.7	4211.4	**4125**	150	77.5	**4125**
Average						5173.0	96.33%	4982.1	3.8	96.47%	4989.2	99.75%	5159.02	80.6	5205.3	**5173.0**	150	44.3	**5173.0**
Set3																			
n22-k4-s13-14	21	2	3	4	4	6396*	95.95%	6137.2	3.8	95.95%	6137.2	99.77%	6381.23	795.2	6406.8	**6396**	150	15.8	**6396**
n22-k4-s13-16	21	2	3	4	4	6922*	97.00%	6714.0	2.1	97.00%	6714.0	99.86%	6912.52	515.1	6954.2	**6922**	150	31.5	**6922**
n22-k4-s13-17	21	2	3	4	4	6408*	93.57%	5996.2	4.1	93.57%	5996.2	98.62%	6319.83	421.1	**6408.0**	**6408**	150	35.4	**6408**
n22-k4-s14-19	21	2	3	4	4	6634	95.27%	6320.4	2.9	95.27%	6320.4	98.60%	6541.32	11995.6	7018.4	**6634**	150	132.1	**6634**
n22-k4-s17-19	21	2	3	4	4	6947*	95.79%	6654.2	4.2	95.79%	6654.2	98.24%	6824.66	20575.4	7094.6	6965	150	72.4	6965
n22-k4-s19-21	21	2	3	4	4	6529*	96.60%	6307.3	8.2	96.67%	6311.6	99.47%	6494.46	8499.0	6625.2	**6529**	150	104.9	**6529**
Average						6639.3	95.70%	6354.9	4.2	95.71%	6355.6	99.09%	6579.00	7133.6	6751.2	6642.3	150	65.4	6642.3

Table 17: Performance analysis on medium-scale instances of Set 2, 3 and 6a

	Characteristics					Lower Bounds							LNS-E2E				
Instance	n_c	n_s	m^1	m^2	n_r	UB	%LB0	LB0	$\mathrm{T_{LB0}}$(s)	%LB1	LB1	$\mathrm{T_{LB1}}$(s)	Avg	Best	BKS	T(s)	T*(s)
Set 2																	
n33-k4-s1-9	32	2	3	4	5	7617	98.46%	7499.4	73.3	98.46%	7499.4	149.4	7751.0	**7617**	**7617**	150	75.3
n33-k4-s2-13	32	2	3	4	5	7925	94.81%	7513.4	44.7	94.81%	7513.4	112.7	8025.0	**7925**	**7925**	150	103.3
n33-k4-s3-17	32	2	3	4	5	8090	92.88%	7514.2	69.7	92.88%	7514.2	181.3	8280.2	**8090**	**8090**	150	107.0
n33-k4-s4-5	32	2	3	4	5	8870	93.84%	8323.8	79.2	93.84%	8323.8	257.1	8925.2	**8870**	**8870**	150	91.0
n33-k4-s7-25	32	2	3	4	5	8318	95.51%	7944.1	77.0	95.74%	7963.3	168.9	8374.8	**8318**	**8318**	150	92.7
n33-k4-s14-22	32	2	3	4	5	8621	98.42%	8484.4	218.5	98.42%	8484.4	529.5	8680.4	**8621**	**8621**	150	90.2
Average						8240.2	95.65%	7879.9	93.7	95.69%	7883.1	233.1	8339.4	**8240.2**	**8240.2**	150	93.3
Set 3																	
n33-k4-s16-22	32	2	3	4	6	7561	91.60%	6926.2	89.4	91.60%	6926.2	328.4	7656.2	**7561**	**7561**	150	94.5
n33-k4-s16-24	32	2	3	4	6	7501	94.77%	7108.5	168.4	94.77%	7108.8	607.1	7520.0	**7501**	**7501**	150	102.7
n33-k4-s19-26	32	2	3	4	6	7212	94.42%	6809.5	98.6	94.42%	6809.5	253.4	7223.2	**7212**	**7212**	150	47.3
n33-k4-s22-26	32	2	3	4	6	7334	95.81%	7027.0	290.5	96.85%	7103.1	738.5	7498.4	**7334**	**7334**	150	131.3
n33-k4-s24-28	32	2	3	4	6	7443	95.40%	7100.5	234.2	96.80%	7204.6	569.9	7371.6	**7443**	**7443**	150	116.2
n33-k4-s25-28	32	2	3	4	6	7429	93.68%	6959.7	258.4	93.68%	6959.7	579.8	7490.4	**7429**	**7429**	150	108.3
Average						7413.3	94.28%	6988.6	189.9	94.69%	7018.6	512.9	7460.0	**7413.3**	**7413.3**	150	100.0
Set 6a																	
A-n51-4	50	4	2	50	5	7663	95.27%	7300.8	121.9	98.76%	7568.0	762.9	7879.4	**7663**	**7663**	150	109.4
A-n51-5	50	5	2	50	6	8268	95.77%	7918.0	98.4	98.16%	8116.0	2783.6	8386.4	**8268**	**8268**	150	64.5
A-n51-6	50	6	2	50	7	7795	93.08%	7255.9	117.1	98.18%	7653.4	15723.7	7943.8	**7795**	**7795**	150	106.0
A-n76-4	75	4	3	75	7	10599	95.40%	10111.7	214.9	97.23%	10305.4	6463.9	10692.0	**10599**	**10599**	150	97.0
A-n76-5	75	5	3	75	7	11178	95.18%	10638.9	175.5	98.17%	10973.6	16406.4	11242.4	**11178**	**11178**	150	88.8
A-n76-6	75	6	3	75	7	10156	95.60%	9709.2	242.2	98.92%	10046.1	85538.4	10250.0	**10156**	**10156**	150	110.7
B-n51-4	50	4	2	50	5	6589	97.20%	6404.4	163.2	97.96%	6454.8	342.4	6791.2	**6589**	**6589**	150	111.6
B-n51-5	50	5	2	50	6	7252	94.73%	6869.8	116.3	95.53%	6928.0	1859.6	7446.4	**7252**	7240	150	90.8
B-n51-6	50	6	2	50	7	6583	95.02%	6255.0	256.0	97.51%	6419.3	3054.1	6787.6	**6583**	**6583**	150	61.4
B-n76-4	75	4	3	75	7	9945	95.99%	9546.7	198.4	98.02%	9748.0	2184.4	9995.8	**9945**	9943	150	99.5
B-n76-5	75	5	3	75	7	9139	94.70%	8655.1	210.4	98.26%	8980.0	9903.7	9209.2	**9139**	**9139**	150	71.9
B-n76-6	75	6	3	75	7	8238	94.44%	7780.1	427.0	97.80%	8056.5	79962.3	8287.6	**8238**	**8238**	150	82.4
C-n51-4	50	4	2	50	5	8407	94.57%	7950.2	137.0	95.74%	8048.5	888.4	8596.2	**8407**	**8407**	150	80.4
C-n51-5	50	5	2	50	6	8810	94.99%	8368.3	261.1	95.77%	8437.3	1346.5	9276.0	**8810**	**8810**	150	82.4
C-n51-6	50	6	2	50	7	8160	93.73%	7648.6	180.3	95.83%	7819.7	7092.7	8390.6	**8160**	**8160**	150	72.9
C-n76-4	75	4	3	75	7	12162	94.61%	11506.7	199.0	98.30%	11955.7	3996.1	12381.2	**12162**	12147	150	99.3
C-n76-5	75	5	3	75	7	13033	92.00%	11990.5	402.2	93.33%	12163.4	38723.3	13247.0	**13033**	**13033**	150	79.3
C-n76-6	75	6	3	75	7	11808	93.28%	11014.5	285.0	97.11%	11466.6	93643.4	12129.8	**11808**	11806	150	92.8
Average						9210.3	94.75%	8718.0	211.4	97.25%	8952.2	20593.1	9385.1	**9210.3**	9208.6	150	88.9

Table 18: Performance analysis on the large-scale instances of Set 5 – Evaluation of the benefits of an integrated routing and charging-stations optimization

						2EVRP						E2EVRP						
	Characteristics					LNS-2E			LNS-E2E ∞			LNS-2E post			LNS-E2E			
Instance	n_c	n_s	m^1	m^2	n_r	Avg	Best	T*(s)	Avg	Best	T*(s)	Avg	Best	T*(s)	Avg	Best	T*(s)	BKS
100-5-1	100	5	5	32	10	15692.6	15640	200.3	15660.9	15639	229.8	16689.9	16593	200.4	16224.6	16167	403.7	16165
100-5-1b	100	5	5	15	10	11124.9	11082	405.9	11208.3	11118	215.7	12802.2	12495	406.0	12070.2	11937	278.4	11937
100-5-2	100	5	5	32	10	10171.0	10157	393.3	10185.1	10157	405.2	11282.4	11189	393.4	10578.0	10578	32.9	10578
100-5-2b	100	5	5	15	10	7814.0	7814	221.5	8032.0	7833	165.4	8833.7	8657	221.6	8426.1	8307	429.4	8307
100-5-3	100	5	5	30	10	10451.0	10451	77.5	10458.1	10451	213.9	10876.7	10863	77.6	10651.4	10651	267.7	10651
100-5-3b	100	5	5	16	10	8283.0	8283	119.5	8287.1	8283	167.7	9341.6	9332	119.6	9068.0	9063	244.8	9018
100-10-1	100	10	5	35	11	11297.0	11247	424.2	11268.2	11247	129.5	11963.8	11942	424.3	11451.8	11409	435.4	11409
100-10-1b	100	10	5	18	11	9243.6	9151	549.3	9257.2	9242	163.3	10374.0	10346	549.4	10194.3	10168	239.5	10168
100-10-2	100	10	5	33	11	10140.8	10100	336.5	10158.0	10127	336.6	10879.0	10829	336.6	10561.5	10525	395.9	10515
100-10-2b	100	10	5	18	11	7825.6	7781	343.4	7965.2	7956	257.5	9558.3	9463	343.5	8800.7	8752	284.0	8621
100-10-3	100	10	5	32	11	10546.9	10503	318.4	10530.1	10490	180.6	11256.4	11164	318.5	10743.6	10730	276.3	10730
100-10-3b	100	10	5	17	11	8646.2	8554	448.7	8683.3	8628	269.2	9503.2	9414	448.8	9209.6	9144	245.2	9144
200-10-1	200	10	5	62	20	15918.6	15615	796.0	15544.6	15453	362.5	16696.7	16426	796.1	16354.8	16016	417.1	16016
200-10-1b	200	10	5	30	20	12310.7	11871	730.7	12076.1	11908	338.5	13510.3	13229	730.8	12975.4	12771	571.8	12768
200-10-2	200	10	5	63	20	13986.3	13669	635.4	13696.4	13577	263.7	14087.1	13971	635.5	14132.9	13860	329.2	13860
200-10-2b	200	10	5	30	20	10089.7	10025	504.2	10257.7	10004	363.0	11099.6	10837	504.3	10833.0	10515	355.4	10495
200-10-3	200	10	5	63	20	18102.1	18000	580.8	17990.0	17925	622.6	18251.8	18167	580.9	18144.4	18094	443.9	18073
200-10-3b	200	10	5	30	20	12088.4	12021	500.4	12001.3	11955	380.0	12711.4	12586	500.5	12515.3	12445	468.0	12441
Average						11318.5	11220.2	421.4	11292.2	11221.8	281.4	12206.6	12083.5	421.5	11829.8	11729.6	339.9	11716.4

Impact of the main LNS components. We conducted additional experiments to measure the contribution of each operator of the LNS-E2E. Starting from the current algorithm (baseline configuration), we deactivated one separate destroy operator listed in Section 3.5.1, in turn, and measured the solution quality of resulting algorithm. In the specific case of the configuration "No open", all candidate satellites are made available again (re-opened) at each restart, instead of using this component as a destroy operator. These experiments were conducted on the 18 large instances of Set 5, using 10 independent runs and a time limit $T_{\mathrm{MAX}} = 15$ minutes. The solution quality is reported as an average percentage gap from the baseline. Table 19 summarizes the results.

Table 19: Sensitivity analysis on the contribution of each operator.

Baseline	A) No related	B) No route	C) No close	D) No open	E) No single
11829.8	2.70%	2.44%	1.52%	3.10%	1.61%

From these results, we first observe that the "open all satellites" operator (D) is essential for the performance of the method, as it allows to control the frequency of the exploration of different satellite configurations. Without the explicit use of a dedicated operator to re-open satellites, the solutions are 3.10% worse on average. The operator "closes satellite" (C) has a smaller but still very significant impact on the overall solution quality, with a deviation of 1.52% from the baseline when deactivated. Therefore, forcing the elimination of some satellite choices at different phases of the method is essential to evaluate structurally different solutions which would not be attained otherwise by the cheapest insertion repair heuristic.

No related measures the deterioration due to the deactivation of the related nodes destruction operator (A), which destroys specific areas around a seed customer. Analogously, *no route* measures the performance deterioration when the operator targeting random routes (B) is deactivated, and column *no single* shows the impact of not using the destruction operator which removes single-customer routes (E). All these operators appear to contribute significantly to the performance of the method, and the deactivation of any of these elements leads to an overall drop of method performance.

We finally tested a version of the method without a restart process after each i_{max} iterations without improvement. In this configuration, the loop of Algorithm 1, Line 4–8 is executed until reaching a maximum time of T_{MAX}. We observed a deterioration of solution quality of 1.90% with respect to the baseline, demonstrating again the importance of diversification components, in this case the restarts mechanism, for the E2E-VRP.

3.6.2 Sensitivity analysis – Density of charging stations and battery capacity

Our second set of experiments focuses on the impact of two defining features of battery-powered distribution networks: the density of the charging stations, and the range of the vehicles. To that extent, we created two additional sets of instances with $n_c = 50$ customers and $n_s = 4$ satellites each, approximating as closely as possible real delivery conditions in a metropolitan area while pertaining to the E2EVRPs class. Set 7 contains 10 groups of 20 instances with a number of charging stations $n_r \in \{2, 3, 5, 10, 15, 20, 25, 30, 40, 50\}$ (in addition to the satellites) and a battery capacity $L = 1000$, whereas Set 8 contains 10 groups of 20 instances with $n_r = 20$ charging stations and a battery capacity $L \in \{800, 900, 1000, \ldots, 1700\}$. When varying the number of charging stations or the battery capacity, all other instance characteristics (satellite locations, customer locations and demands as well as the existing charging station locations) are kept identical.

In each instance, 40 customers have been located randomly (with uniform probability) in an ellipse X_1 centered in (1000,500), with x-axis of dimension 800 and y-axis of dimension 400, and 10 additional customers have been located randomly in an ellipse X_2 with the same center, an x-axis of dimension 1000 and y-axis of dimension 500. The locations of the satellites are picked randomly in the area formed by $X_2 - X_1$, and the depot is fixed in position (300,0). Moreover, 80% of the charging stations are randomly located in X_1, and 20% in X_2. The demand quantity of each customer is randomly selected in [1,25]. Six 1^{st}-level vehicles with capacity $Q_1 = 250$ are available, and ten 2^{nd}-level electric vehicle with capacity $Q_2 = 125$ are available at each satellite.

Considering that one distance unit in each instance corresponds to 0.1km on a map, the area considered for the location of customers and charging stations covers 15708 km^2, a size similar in magnitude with the metropolitan area of Paris. We set a baseline of $L = 1000$ for the battery capacity, equivalent to a range of 100km. This value matches the specifications of the Renault Kangoo Zoe and Nissan Leaf 2015/2016 minus a safety range of 30km. Varying this parameter allows to evaluate the impact of the battery capacity.

Figure 5 depicts the evolution of the operational costs as a function of the number of charging stations n_r, and Figure 6 shows the impact of the battery capacity L. The results are expressed as percentage gaps between the cost of the LNS-E2E solution with and without battery restrictions (i.e. percentage detour miles due to recharging), and averaged over all 20 instances of each class. The average number of visits to charging stations in the solutions is also represented on each graph.

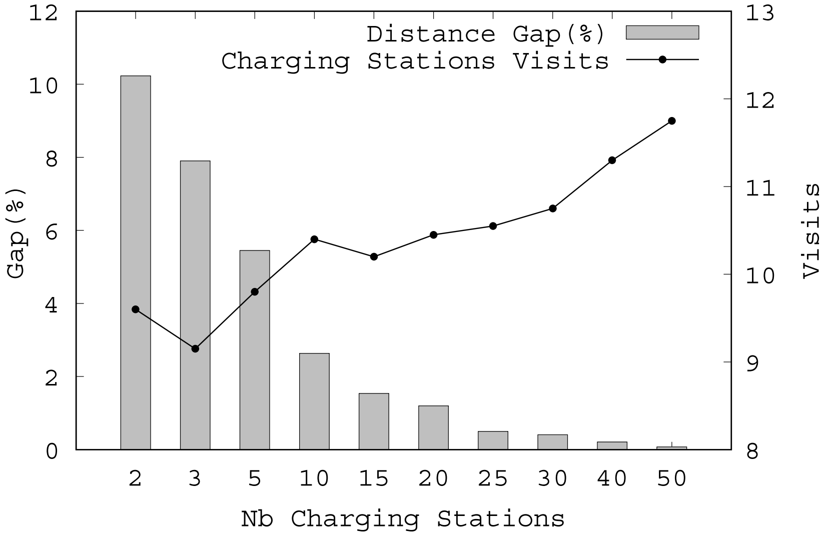


Figure 5: Impact of the number of available charging stations on the detour costs due to recharging and the number of visits to stations.

These experiments highlight the significant impact of the charging stations density and vehicles batteries capacities in the instances under study. As the number of charging stations grows, the detour costs due to recharging stations visits rapidly decreases: e.g., from 5.45% in average when $n_r = 5$ to 1.53% when $n_r = 15$. Conducting a power-law regression of the form $f(n_r) = \alpha/n_r^\beta$ (least-squares regression of an affine function on the log-log graph), the extra detours due to recharging diminish proportionally to $1/\rho^{1.24}$. In these conditions, doubling the number of charging stations allows to reduce extra recharging costs by approximately 58%.

Interestingly, the *number of visits* to charging stations slightly increases with n_r: from 9.5 in average when $n_r = 2$ to 11.75 when $n_r = 50$. Indeed, when the recharging station network is sparse, most detour options to recharging stations involve significant extra costs, and the vehicle routing algorithm tends to reduce to a strict minimum the number of such detours. In contrast, in the presence of a dense recharging stations network, the solutions converge more closely towards the 2E-VRP cost (disregarding battery constraints) as there are always multiple options of charging stations on the way.

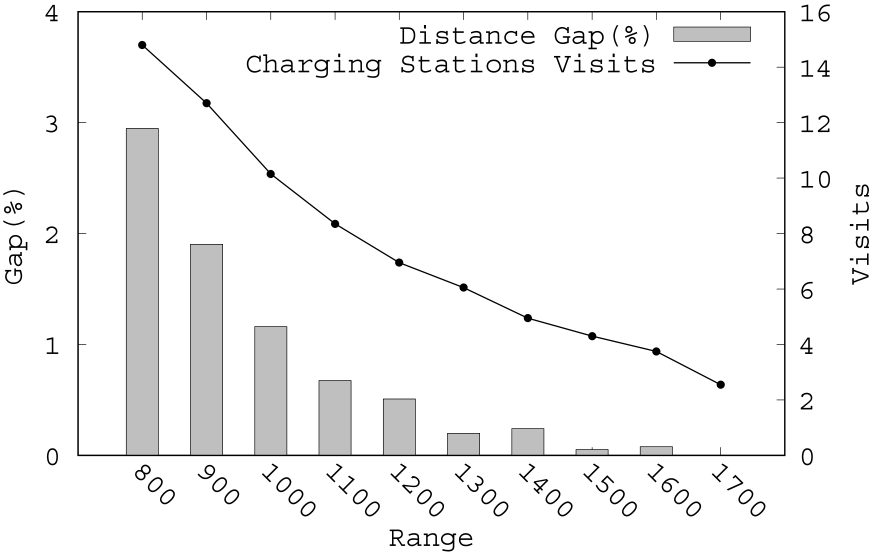


Figure 6: Impact of the vehicle range (i.e., battery capacity) on the detour costs due to recharging and the number of visits to stations.

The vehicles' range (i.e., battery capacity), has an even larger impact (see Figure 6). For most of the considered instances, a range below 700 distance units (= 70km) would lead to an infeasible problem, as it becomes impossible to travel between some pairs of customers and find adequate recharging locations en-route. Therefore, adequate battery technology is a key factor for the viability of battery-powered delivery networks. The extra costs due to recharging and number of visits to recharging stations tend to be high when considering vehicles' ranges close to the feasible limit ($L = 800$). These values then rapidly decrease to become close to zero when the range L exceeds 1500 (i.e., 150km), a value which may be soon attained by lightweight electric trucks. Once this regime is attained, the battery capacity becomes sufficient to do most tours without en-route recharging visits. Still, manufacturing the current best-performing batteries on a global scale requires a large supply of minerals (e.g., nickel and cobalt) which are only accessible in limited quantities in the environment. As illustrated in Figure 5, the development of good fast-charging infrastructures is another strategic development path to obtain operational efficiency, which may turn out, in the long run, to be more sustainable and economical than a race towards heavier and more robust batteries.

3.7 Conclusions

In this paper, we have formulated the E2EVRP, an extension of the 2EVRP involving electric vehicles for second-echelon deliveries, battery capacity constraints, and possible visits to charging stations, and used it as a prototypical problem for the study of multi-echelon battery-powered supply chains. We introduced an efficient exact algorithm, based on the enumeration of candidate solutions for the first echelon and on bounding functions and route enumeration for the second echelon, along with a problem-tailored large neighborhood search metaheuristic (LNS-E2E). A comparison of the solutions found by the LNS-E2E with lower bounds and optimal solutions produced by the mathematical programming algorithm demonstrates the excellent performance of both algorithms. In particular, all known optimal solutions for small instances were retrieved by the LNS-E2E, and an average optimality gap of 3.57% between the best known upper and lower bounds was obtained on medium-scale instances. The metaheuristic was also evaluated on the classical 2EVRP (without electric vehicles), producing new state-of-the-art solutions on large-scale instances with 200 customers. Finally, thanks to the use of efficient heuristic move filters in the local search and labeling algorithms, the computational effort of LNS-E2E remains comparable with that of previous metaheuristics for the classical 2EVRP.

Beyond the usefulness of these optimization algorithms for the operational planning of electric fleets, this paper brought new managerial insights related to the incorporation of electric vehicles into two-echelon delivery networks and to the recharging-stations infrastructure required for an efficient supply chain. For this additional study, we created 400 additional test instances simulating typical requests patterns and delivery infrastructures in a metropolitan area with varying density of charging stations vehicles' battery capacities. We observed that the detour miles due to recharging decrease in $\mathcal{O}(1/\rho^x)$ with $x \approx 5/4$ as a function of the number of charging stations. Moreover, the range of the electric vehicles has an even bigger impact: an increase of battery capacity to a range of 150km helps performing the majority of suburban delivery tours without need for en-route recharging, but a battery capacity below 80km render electric deliveries unviable in our setting. Between these two extremes, the extra costs due to recharging quickly decrease as a function of the battery capacity.

The future research perspectives are multiple. Firstly, we recommend to pursue the study of exact methods and metaheuristics for multi-echelon electric VRPs. These optimization problems involve a large number of decision classes, related to satellite choices, recharging stations choices, and vehicle routing at two levels, posing a formidable challenge for exact and heuristic algorithms alike. With the rapid development of battery-powered vehicles and green supply chains, efficient algorithms for large scale problems are critically needed, but the current methods still need to be improved in terms of accuracy, scalability, and generality, e.g., considering possible extensions to multi-echelon electric delivery schemes arising in city logistics (Cattaruzza et al., 2017), other vehicle routing attributes (Vidal et al., 2013) and stochastic settings.

Secondly, our sensitivity analyses on electric vehicles characteristics and other strategic decisions (number and placement of charging stations) could be extended further. One limitation of the current study relates to the placement of the charging stations, which is randomized in a fixed area. However, during urban planning, recharging stations are placed in strategic locations to meet the needs of the population, or based on competitive location approaches. Solving this strategic location optimization problem may be necessary for a more accurate sensitivity analysis. Yet, it is an intricate problem, which can be viewed as a variant of location routing problems (Schiffer and Walther, 2017a,b), or modeled as a bilevel optimization problem and congestion game (Xiong et al., 2015). To this date, the optimization of charging stations locations has never been considered in the context of a multi-echelon delivery network, forming a promising research avenue.

3.8 Appendix

3.8.1 Example of Labelling Algorithm

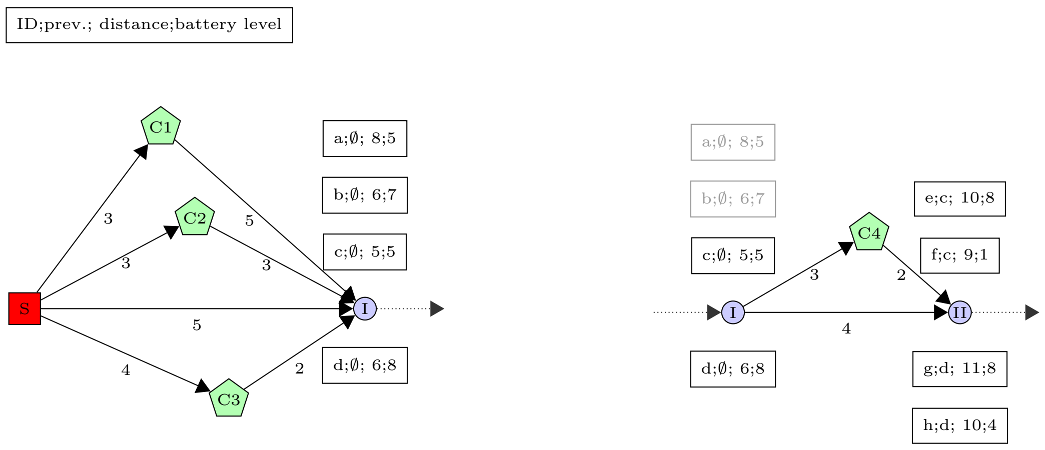
The following example and illustration is adapted from Breunig et al. (2017).

Figure 7 shows the process of labelling. The depicted labels show the cost, as well as the current charging level. We assume a battery capacity of ten units. For simplicity, driving one unit of distance increases the driving distance by one unit and depletes the battery level by one unit. Starting from the home satellite S (red square), the vehicle can go to the first customer I directly, or via one of three charging stations (green pentagons *C1*-*C3*), shown in Figure 7a. Label a at node I stores the driving cost of eight (S to *C1*: 3 + *C1* to I: 5) and a battery level of five. Starting at full capacity at S, the charging station can be reached at a level of seven; the vehicle then fully charges up to ten again and after going five more units reaches customer I at a level of five. The next label corresponds to the route *S-C2-I*, below is label c for going directly from S to I without recharging. If the vehicle travels through *C3* it arrives at I with a cost of seven and a charging level of eight (Label d).

Dominated labels can now be eliminated. Label a is dominated by c as the vehicle can reach customer I at the same charging level but at a lower cost. Label b is dominated by d as I can be reached at the same cost of six, but at a higher charging level. The remaining labels are then extended to the next node on the route: customer *II* in Figure 7b. The vehicle can travel from I to *II* directly, or via charging station *C4*. The two upper labels at node *II* evaluate these options starting from label c, the lower two labels work in a similar way when extending label d. Again, some labels can then be eliminated: g and h are both dominated by e.

Acknowledgements

The authors would like to thank the anonymous reviewers and associate editor for their helpful suggestions. This study was partially funded by the Austrian Climate and Energy Fund within the “Electric Mobility Flagship projects” program under grant 834868 (project VECEPT) in Austria, and CNPq and FAPERJ in Brazil (grant numbers 308498/2015-1 and E-26/203.310/2016). This support is gratefully acknowledged.



(a) S to I

(b) I to II

Figure 7: Illustration of the labelling algorithm

4 Recent publications in this area of research

We will cover the recent works from scientific literature which also dealt with the before mentioned two-echelon problems or similar ones.

Amarouche et al. (2018) developed a hybrid heuristic for the classic 2EVRP. In a first phase, a neighbourhood search (destroy, repair and local search) generates promising routes. Those routes are then recombined in a set covering problem. Their algorithm was able to slightly improve seven best known solutions on instances of the large Set 5, and also performs slightly better than the LNS from Breunig et al. (2016). Over all benchmark instance sets, which were used in Breunig et al. (2016) they were able to close the gaps from 0.18% (LNS) to 0.10% (hybrid heuristic) for the average solution runs, respectively from 0.09% to 0.04% for the best found solutions. The calculation of their runtimes is not entirely clear. Although they state that they set the stopping criterion to 60s for smaller instances, and to 900s for larger instances - as done in Breunig et al. (2016, 2019), they report way shorter CPU times in the tables - where we can only guess, that this might be the time when the best solution was found. Unfortunately a request to clarify this issue was not responded to.

Dellaert et al. (2019) introduced new instances with time windows at customers. They developed an exact method and were able to solve their instances with up to 100 customers to optimality; unfortunately they did not apply their algorithm to existing instances without time windows for comparison to previous literature.

Jie et al. (2019) built upon Breunig et al. (2016) and extended the 2EVRP to another electric variant, using battery swapping stations, and electric vehicles on both echelons. They also tested their hybrid algorithm, combining column generation and an ALNS, on the classic 2EVRP instances. Their algorithm generated solutions not necessarily better, but similar to the results in Breunig et al. (2016).

Marques et al. (2020) developed a branch-price-and-cut technique, and were able to improve three of the best known solutions from Breunig et al. (2016) of Set 6 instances with 100 customers, on average by 0.07%.

Hof and Schneider (2021) introduced the vehicle routing problem with time-windows and mobile depots, which is very similar to 2EVRP, but all vehicles start from the same depot, and small vehicles meet the large vehicles again on their routes - which they call intraroute resource replenishments.

Mhamedi et al. (2021) applied their branch-price-and-cut method to the two-echelon vehicle routing problem with time windows (2EVRPTW), improving the results of Dellaert et al. (2019) and solving some more instances to optimality.

Anderluh et al. (2021) extended the problem by several objective functions. They consider economic, environmental and social objectives and transform the problem to a multi-objective one, which consecutively they solve with a sophisticated LNS, embedded in a heuristic rectangle/cuboid splitting framework to deal with the different objectives.

An extensive survey over different two-echelon related problems was just published online by Sluijk et al. (2022).

5 Conclusion

Working on variants of two-echelon routing problems we could show the specific strengths of this set-up for the delivery (or pickup) of goods in current city logistics. Using different vehicles for different purposes can have advantages on daily life in urban areas: keeping large vehicles outside the city's boundaries but being able to ship larger quantities at once, and using smaller, potentially battery powered vehicles in densely populated areas.

The proposed solution techniques for different problem settings were proven to be effective and generate high quality solutions within very reasonable time. There are definitely more opportunities for research and to shift those problem settings even closer to real world applications, as is already done by the introduction of time windows at customers, shifting even first-level vehicles to battery powered ones, or taking a multi-objective approach considering several different impacts of transport.

Bibliography

References

Y. Amarouche, R. N. Guibadj, and A. Moukrim. A Neighborhood Search and Set Cover Hybrid Heuristic for the Two-Echelon Vehicle Routing Problem. In R. Borndörfer and S. Storandt, editors, *18th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS 2018)*, volume 65 of *OpenAccess Series in Informatics (OASIcs)*, pages 11:1–11:15, Dagstuhl, Germany, 2018. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

A. Anderluh, P. C. Nolz, V. C. Hemmelmayr, and T. G. Crainic. Multi-objective optimization of a two-echelon vehicle routing problem with vehicle synchronization and grey zone customers arising in urban logistics. *European Journal of Operational Research*, 289(3):940–958, 2021.

R. Baldacci. personal communication. email, May 20th 2013.

R. Baldacci and A. Mingozzi. A unified exact method for solving different classes of vehicle routing problems. *Mathematical Programming Ser. A*, 120(2):347–380, 2009.

R. Baldacci, A. Mingozzi, and R. Roberti. New route relaxation and pricing strategies for the vehicle routing problem. *Operations Research*, 59(5):1269–1283, 2011.

R. Baldacci, A. Mingozzi, R. Roberti, and R. Wolfler Calvo. An exact algorithm for the two-echelon capacitated vehicle routing problem. *Operations Research*, 61(2):298–314, 2013.

J. E. Beasley. OR-Library, accessed Oct. 6th 2014. URL `http://people.brunel.ac.uk/~mastjjb/jeb/orlib/files/vrp2e.rar`.

M. Boccia, T. G. Crainic, A. Sforza, and C. Sterle. Location-Routing Models for Designing a Two-Echelon Freight Distribution System. Technical report, Cirrelt, 2011.

U. Breunig, V. Schmid, R. Hartl, and T. Vidal. A large neighbourhood based heuristic for two-echelon routing problems. *Computers & Operations Research*, 76:208–225, 2016.

U. Breunig, R. F. Hartl, and T. Vidal. The two-echelon vehicle routing problem with electric vehicles. In *Anais do XLIX - Simpósio Brasileiro de Pesquisa Operacional*, pages 3797–3807, Blumenau/SC, 2017. SOBRAPO – Sociedade Brasileira Pesquisa Operacional, Rio de Janeiro/RJ.

U. Breunig, R. Baldacci, R. F. Hartl, and T. Vidal. The electric two-echelon vehicle routing problem. *Computers and Operations Research*, 103(March):198–210, 2019.

D. Cattaruzza, N. Absi, D. Feillet, and J. González-Feliu. Vehicle routing problems for city logistics. *EURO Journal on Transportation and Logistics*, 6(1):51–79, 2017.

N. Christofides, A. Mingozzi, and P. Toth. Exact algorithms for the vehicle routing problem based on spanning tree and shortest path relaxation. *Mathematical Programming*, 10:255–280, 1981.

R. G. Conrad and M. A. Figliozzi. The recharging vehicle routing problem. In T. Doolen and E. V. Aken, editors, *Proceedings of the 2011 Industrial Engineering Research Conference*, pages 1–8, 2011.

C. Contardo, V. C. Hemmelmayr, and T. G. Crainic. Lower and upper bounds for the two-echelon capacitated location-routing problem. *Computers & Operations Research*, 39: 3185–3199, 2012.

J.-F. Cordeau, M. Gendreau, G. Laporte, J.-Y. Potvin, and F. Semet. A guide to vehicle routing heuristics. *Journal of the Operational Research Society*, 53(5):512–522, 2002.

T. G. Crainic, N. Ricciardi, and G. Storchi. Advanced Freight Transportation Systems for Congested Urban Areas. *Transportation Research Part C: Emerging Technologies*, 12(2): 119–137, 2004.

T. G. Crainic, N. Ricciardi, and G. Storchi. Models for Evaluating and Planning City Logistics Systems. *Transportation Science*, 43(4):432–454, 2009.

T. G. Crainic, G. Perboli, S. Mancini, and R. Tadei. Two-Echelon Vehicle Routing Problem: a Satellite Location Analysis. *Procedia - Social and Behavioral Sciences*, 2(3):5944–5955, 2010.

T. G. Crainic, S. Mancini, G. Perboli, and R. Tadei. Multi-Start Heuristics for the Two-Echelon Vehicle Routing Problem. In P. Merz and J.-K. Hao, editors, *Evolutionary Computation in Combinatorial Optimization*, volume 6622, pages 179–190. Springer Berlin Heidelberg, Berlin Heidelberg, 2011.

G. A. Croes. A method for solving traveling-salesman problems. *Operations Research*, 6(6): 791–812, 1958.

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

N. Dellaert, F. Dashty Saridarq, T. Van Woensel, and T. G. Crainic. Branch-and-price based algorithms for the two-echelon vehicle routing problem with time windows. *Transportation Science*, 53(2):463–479, 2019.

G. Desaulniers, F. Errico, S. Irnich, and M. Schneider. Exact algorithms for electric vehicle-routing problems with time windows. *Operations Research*, 64(6):1388–1405, 2016.

S. Erdogan and E. Miller-Hooks. A green vehicle routing problem. *Transportation Research Part E: Logistics and Transportation Review*, 48(1):100–114, 2012.

EV World. Beijing readies electric buses for summer olympics. `http://www.evworld.com/news.cfm?newsid=18730` (accessed Oct 13, 2018), 2008.

A. Felipe, M. T. Ortuno, G. Righini, and G. Tirado. A heuristic approach for the green vehicle routing problem with multiple technologies and partial recharges. *Transportation Research Part E: Logistics and Transportation Review*, 71:111–128, 2014.

M. Foltyński. Electric fleets in urban logistics. *Procedia – Social and Behavioral Sciences*, 151: 48–59, 2014.

D. Goeke and M. Schneider. Routing a mixed fleet of electric and conventional vehicles. *European Journal of Operational Research*, 245(1):81–99, 2015.

J. Gonzalez-Feliu, G. Perboli, R. Tadei, and D. Vigo. The two-echelon capacitated vehicle routing problem. Technical report, DEIS OR.INGCE 2007/2(R), University of Bologna, Bologna, Italy, 2008.

N. Hansen. The CMA evolution strategy: a comparing review. In J. Lozano, P. Larranaga, I. Inza, and E. Bengoetxea, editors, *Towards a new evolutionary computation. Advances on estimation of distribution algorithms*, pages 75–102. Springer Berlin Heidelberg, 2006.

V. C. Hemmelmayr. personal communication. email, March 21st 2013.

V. C. Hemmelmayr, J.-F. Cordeau, and T. G. Crainic. An adaptive large neighborhood search heuristic for two-echelon vehicle routing problems arising in city logistics. *Computers & Operations Research*, 39(12):3215–3228, 2012.

G. Hiermann, J. Puchinger, S. Ropke, and R. F. Hartl. The electric fleet size and mix vehicle routing problem with time windows and recharging stations. *European Journal of Operational Research*, 252(3):995–1018, 2016.

G. Hiermann, R. Hartl, J. Puchinger, and T. Vidal. Routing a mix of conventional, plug-in hybrid, and electric vehicles. *European Journal of Operational Research*, 272(1):235–248, 2019.

J. Hof and M. Schneider. Intraroute resource replenishment with mobile depots. *Transportation Science*, 55(3):660–686, May 2021.

J. Hof, M. Schneider, and D. Goeke. Solving the battery swap station location-routing problem with capacitated electric vehicles using an AVNS algorithm for vehicle-routing problems with intermediate stops. *Transportation Research Part B: Methodological*, 97:102–112, 2017.

S. Jacobsen and O. Madsen. A Comparative Study of Heuristics for a Two-Level Routing-Location Problem. *European Journal of Operational Research*, 5(6):378–387, 1980.

JD.com. Urban truck routing and scheduling contest page. `http://jdata.joybuy.com/en/html/detail.html?id=5` (accessed Oct 13, 2018), 2018.

M. Jepsen, S. Spoorendonk, and S. Ropke. A branch-and-cut algorithm for the symmetric two-echelon capacitated vehicle routing problem. *Transportation Science*, 47(1):23–37, 2012.

W. Jie, J. Yang, M. Zhang, and Y. Huang. The two-echelon capacitated electric vehicle routing problem with battery swapping stations: Formulation and efficient methodology. *European Journal of Operational Research*, 272(3):879–904, 2019.

J. John. JD.com CEO sets february 2018 target to change its fleet of delivery vans in beijing to electric. `https://www.gizmochina.com/2017/12/13/jd-com` (accessed Oct 13, 2018), 2017.

M. Keskin and B. Çatay. A matheuristic method for the electric vehicle routing problem with time windows and fast chargers. Technical report, Sabanci University, Istanbul, Turkey, 2018.

M. Keskin and B. Çatay. Partial recharge strategies for the electric vehicle routing problem with time windows. *Transportation Research Part C: Emerging Technologies*, 65:111–127, 2016.

G. Laporte. Location routing problems. In B. L. Golden and A. A. Assad, editors, *Vehicle Routing: Methods and Studies*, pages 163–197. Elsevier Science Publishers B.V., North-Holland, 1988.

G. Laporte and Y. Nobert. A vehicle flow model for the optimal design of a two-echelon distribution system. In H. A. Eiselt and G. Pederzoli, editors, *Advances in Optimization and Control: Proceedings of the Conference "Optimization Days 86" Held at Montreal, Canada, April 30 – May 2, 1986*, pages 158–173, Berlin, Heidelberg, 1988. Springer Berlin Heidelberg.

O. B. Madsen. Methods for solving combined two level location-routing problems of realistic dimensions. *European Journal of Operational Research*, 12(3):295–301, mar 1983.

G. Marques, R. Sadykov, J.-C. Deschamps, and R. Dupas. An improved branch-cut-and-price algorithm for the two-echelon capacitated vehicle routing problem. *Computers & Operations Research*, 114:104833, 2020.

T. Mhamedi, H. Andersson, M. Cherkesly, and G. Desaulniers. A branch-price-and-cut algorithm for the two-echelon vehicle routing problem with time windows. *Transportation Science*, dec 2021.

J.-A. Montoya. *Electric vehicle routing problems: models and solution approaches.* PhD thesis, Université d'Angers, 2016.

V. P. Nguyen, C. Prins, and C. Prodhon. Solving the two-echelon location routing problem by a GRASP reinforced by a learning process and path relinking. *European Journal of Operational Research*, 216(1):113–126, 2012a.

V. P. Nguyen, C. Prins, and C. Prodhon. 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(1):56–71, 2012b.

S. Pelletier, O. Jabali, and G. Laporte. 50th anniversary invited article – goods distribution with electric vehicles: review and research perspectives. *Transportation Science*, 50(1):3–22, 2016.

G. Perboli and M. Rosano. A decision support system for optimizing the last-mile by mixing traditional and green logistics. In C. Temponi and N. Vandaele, editors, *Information Systems, Logistics, and Supply Chain*, pages 28–46. Springer International Publishing, 2018.

G. Perboli, R. Tadei, and F. Masoero. New families of valid inequalities for the two-echelon vehicle routing problem. *Electronic Notes in Discrete Mathematics*, 36(Supplement C): 639–646, 2010.

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

G. Perboli, M. Rosano, M. Saint-Guillain, and P. Rizzo. Simulationï¿½-optimisation framework for City Logistics: an application on multimodal last-mile delivery. *IET Intelligent Transport Systems*, 12(4):262–269, 2018a.

G. Perboli, R. Tadei, and E. Fadda. New Valid Inequalities for the Two-Echelon Capacitated Vehicle Routing Problem. *Electronic Notes in Discrete Mathematics*, 64:75–84, 2018b.

S. Pirkwieser and G. R. Raidl. Variable Neighborhood Search coupled with ILP-based Very Large Neighborhood Searches for the (Periodic) Location-Routing Problem. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 6373 LNCS, pages 174–189, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

D. Pisinger and S. Ropke. A general heuristic for vehicle routing problems. *Computers & Operations Research*, 34(8):2403–2435, 2007.

D. Pisinger and S. Ropke. Large Neighborhood Search. In M. Gendreau and J.-Y. Potvin, editors, *Handbook of Metaheuristics*, volume 146, pages 399–419. Springer, 2010.

S. Poikonen, X. Wang, and B. Golden. The vehicle routing problem with drones: Extended models and connections. *Networks*, 70(1):34–43, 2017.

F. A. Santos, A. Salles da Cunha, and G. R. Mateus. Branch-and-price algorithms for the Two-Echelon Capacitated Vehicle Routing Problem. *Optimization Letters*, 7:1537–1547, 2013.

F. A. Santos, G. R. Mateus, and A. Salles da Cunha. A branch-and-cut-and-price algorithm for the two-echelon capacitated vehicle routing problem. *Transportation Science*, 49(2):355–368, May 2015.

M. Schiffer and G. Walther. The electric location routing problem with time windows and partial recharging. *European Journal of Operational Research*, 260(3):995–1013, 2017a.

M. Schiffer and G. Walther. An adaptive large neighborhood search for the location-routing problem with intra-route facilities. *Transportation Science, Articles in Advance*, 2017b.

M. Schiffer, S. Stütz, and G. Walther. Are ECVs breaking even? - Competitiveness of electric commercial vehicles in medium-duty logistics networks. Technical report, OM-02, 2016.

M. Schiffer, M. Schneider, G. Walther, and G. Laporte. Vehicle routing and location-routing with intermediate stops: A review. *Transportation Science*, 53(2):319–343, 2019.

M. Schneider, A. Stenger, and D. Goeke. The electric vehicle routing problem with time windows and recharging stations. *Transportation Science*, 48(4):500–520, 2014.

G. Schrimpf, J. Schneider, H. Stamm-Wilbrandt, and G. Dueck. Record Breaking Optimization Results Using the Ruin and Recreate Principle. *Journal of Computational Physics*, 159: 139–171, 2000.

M. Schwengerer, S. Pirkwieser, and G. R. Raidl. A variable neighborhood search approach for the two-echelon location-routing problem. In J.-K. Hao and M. Middendorf, editors, *Evolutionary Computation in Combinatorial Optimization*, volume 7245 of *Lecture Notes in Computer Science*, pages 13–24. Springer Berlin Heidelberg, 2012.

P. Shaw. Using constraint programming and local search methods to solve vehicle routing problems. In M. Maher and J.-F. Puget, editors, *Principles and Practice of Constraint Programming*, volume 1520 of *Lecture Notes in Computer Science*, pages 417–431. Springer Berlin Heidelberg, 1998.

N. Sluijk, A. M. Florio, J. Kinable, N. Dellaert, and T. Van Woensel. Two-echelon vehicle routing problems: A literature review. *European Journal of Operational Research*, 2022.

E. Taillard, P. Badeau, M. Gendreau, F. Guertin, and J.-Y. Potvin. A tabu search heuristic for the vehicle routing problem with soft time windows. *Transportation Science*, 31(2):170–186, 1997.

The R Foundation for Statistical Computing. R help documentation on boxplot.stats, version 3.1.2, 2014. URL `http://web.mit.edu/r/r_v3.0.1/lib/R/library/grDevices/html/boxplot.stats.html`. retrieved Feb. 20th 2015.

P. Toth and D. Vigo. The granular tabu search and its application to the vehicle-routing problem. *INFORMS Journal on Computing*, 15(4):333–346, 2003.

T. Vidal, T. Crainic, M. Gendreau, and C. Prins. Heuristics for multi-attribute vehicle routing problems: A survey and synthesis. *European Journal of Operational Research*, 231(1):1–21, 2013.

K. Wang, S. Lan, and Y. Zhao. A genetic-algorithm-based approach to the two-echelon capacitated vehicle routing problem with stochastic demands in logistics service. *Journal of the Operational Research Society*, pages 1–13, 2017a.

X. Wang, S. Poikonen, and B. Golden. The vehicle routing problem with drones: Several worst-case results. *Optimization Letters*, 11(4):679–697, 2017b.

Y. Wang, Y. Huang, J. Xu, and N. Barclay. Optimal recharging scheduling for urban electric buses: A case study in Davis. *Transportation Research Part E: Logistics and Transportation Review*, 100:115–132, 2017c.

Wr. Linien. Wiener Linien mit Staatspreis Mobilität ausgezeichnet. `https://www.wien.gv.at/presse/2013/09/13/wiener-linien-mit-staatspreis-mobilitaet-ausgezeichnet`, in German, (accessed Oct 13, 2018), 2013.

Y. Xiong, J. Gan, B. An, C. Miao, and A. Bazzan. Optimal electric vehicle charging station placement. In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, pages 2662–2668, 2015.

J. Yang and H. Sun. Battery swap station location-routing problem with capacitated electric vehicles. *Computers & Operations Research*, 55:217–232, 2015.

Z.-Y. Zeng, W.-S. Xu, Z.-Y. Xu, and W.-H. Shao. A hybrid GRASP+VND heuristic for the two-echelon vehicle routing problem arising in city logistics. *Mathematical Problems in Engineering*, 2014:1–11, 2014.