The two-echelon vehicle routing problem with electric vehicles

Ulrich Breunig
Department of Business Administration, University of Vienna
Oskar-Morgenstern-Platz 1, 1090 Vienna, Austria
ulrich.breunig@univie.ac.at

Richard F. Hartl
Department of Business Administration, University of Vienna
Oskar-Morgenstern-Platz 1, 1090 Vienna, Austria
richard.hartl@univie.ac.at

Thibaut Vidal
Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro
Rua Marquês de São Vicente, 225 Gávea, Rio de Janeiro - RJ, 22451-900, Brazil
vidalt@inf.puc-rio.br

ABSTRACT

Two-echelon distribution systems provide many advantages for city logistics and help keeping large vehicles out of densely populated city centers. As reflected by numerous studies on the two-echelon vehicle routing problem and the two-echelon location routing problem, large trucks can be efficiently used to deliver goods to intermediate facilities, and smaller vehicles allow to service end customers. These smaller vehicle are designed for services within a limited range; and their size, pollution and noise emissions must be minimized as possible.

In such a context, using electric vehicles for the second echelon is very relevant. Yet, these vehicles have a more limited delivery range due to their small battery capacity compared to conventional vehicles. Moreover, the inclusion of visits to charging stations, which have to be already planned during route generation leads to significant methodological challenges. We thus introduce the electric two-echelon vehicle routing problem as a prototypical problem to study these challenges. To solve it, we propose an hybrid large neighbourhood search heuristic, based on a few efficient destruction and reconstruction operators as well as a strong local-search procedure. During solution creation and improvement, a fast labeling algorithm is used to find the optimal positions for charging stations visits. Our computational experiments on a variety of problem instances demonstrate the good performance of the methods.

KEYWORDS. Two-Echelon Vehicle Routing Problem. Electric Vehicles. Large Neighborhood Search, Local Search, Dynamic Programming

1. Introduction

High population densities, globalization of production, and the widespread request of home deliveries and probably many other factors all demand effective planning of vehicle itineraries for deliveries. Often referred to as city logistics, there have been a plethora of studies focused on different aspects of transport within cities. In city centres, space is limited, and thus noise and pollution affect many people negatively. Therefore it is an aim to optimise ways of delivery while still satisfyingly providing the inhabitant's needs.

Several studies have been dedicated to multi-layered transportation approaches. In such a scheme, the supply chain is divided into different levels, each with distinct types of vehicles. For interurban transportation, one can take advantage of large trucks, while small and quiet vehicles are preferred within city centres. The two-echelon vehicle routing problem (2EVRP), in particular, makes use of such a two-level set-up. As technology for electric mobility advances further, we believe that this delivery scheme is naturally destined to make use of electric vehicles. Yet, one of the major drawbacks is the limited range due to the complexity of storing electric energy. Batteries also tend to be quite heavy, such that the payload for electric vehicles tend to be lower than that of conventional vehicles.

In this paper, we investigate a transportation setting in which electric vehicles are used only for short distance delivery in the inner city. When the planned route is longer than the range of a vehicle, we also allow the use of recharging stations en route. From a combinatorial optimization viewpoint, several challenging classes of decisions have to be taken simultaneously: finding good itineraries for each of the vehicles; decide at which locations goods should be transferred from first level trucks to smaller battery-powered vehicles, and which charging stations should be used.

To address this problem we propose a simple large neighbourhood search (LNS) metaheuristic with local search operators and a labelling algorithm for selecting charging stations.

The paper is organised as follows: Section 2 describes the problem. Section 3 explores the relevant literature on related topics. Section 4 details the proposed solution method. The benchmark instances and the computational experiments are presented in Sections 5–6, and Section 7 concludes.

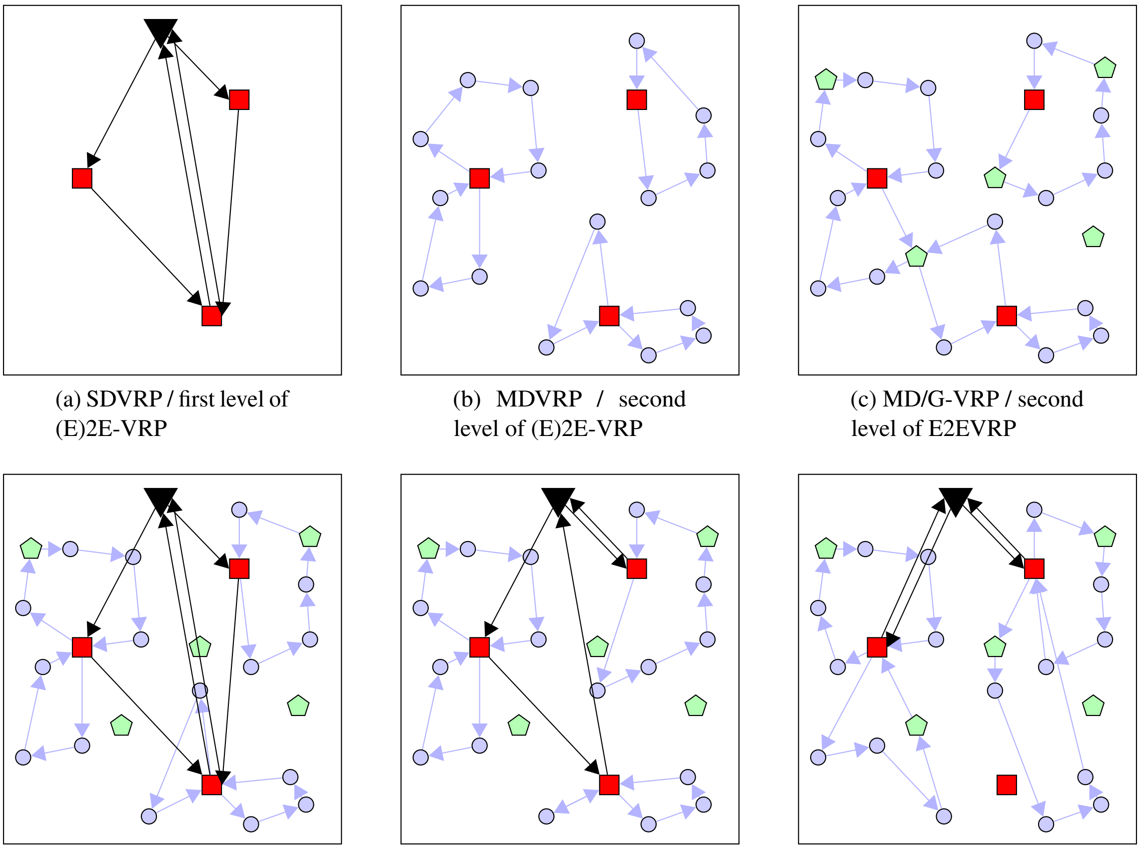
2. Problem Description

We propose the electric two-echelon vehicle routing problem (E2EVRP), which is based on the classic 2EVRP. Several customers demand a known quantity of a homogeneous product. There are two different types of vehicles involved in the distribution system. On the first level, large trucks transport the goods from a central depot or warehouse (black triangle in Figure 1) to several intermediate facilities, called satellites (red squares). At each satellite location several vehicles are available, which are smaller than first level trucks by design. They are often referred to as city freighters. They deliver the goods from satellites to the end customers, forming the second level. Goods have to be transferred from first level trucks to second level vehicles at a satellite, and no direct shipping from the depot to a customer is allowed. Moreover, each customer has to receive all of its demand at once, such that no split deliveries are allowed on the second level. Not all satellites have to be used. The 2EVRP can be decomposed as follows: the first level corresponds to a capacitated vehicle routing problem with split deliveries (CVRPSD) (Figure 1a), the second level corresponds to a multi-depot vehicle routing problem (MDVRP). Both levels are obviously linked, as the correct quantities have to be provided and transferred.

Electric vehicles are particularly adequate for short distance deliveries in city centers. This is why we introduce an extension of the 2EVRP, called the E2EVRP, in which the city freighters are battery powered. If a desired route on the second level is longer than the range of the electric vehicle, then we allow them to be recharged at electric charging stations or satellite locations. Indeed, there is usually sufficient space in a satellite location to park the vehicles overnight, so we assume satellites also provide connection to the electrical grid, i.e. a charging station. Of course, there may be more locations to charge an electric car in a city, so there are also additional charging facility locations (green pentagons in Figure 1). One electric vehicle can be recharged zero, one, or several times on its route (Figure 1c). Charging stations can be used multiple times, also by different vehicles - or not be in use at all. See different feasible solutions for a E2EVRP in Figures 1d to 1f.

The possibilities for city logistics are manifold. For this specific problem type, we do not have to limit ourselves to the use of electric cars —any other means of transport, which has to be recharged en route can be considered. This could be applied to a “carless” city: first level trucks operate deliveries to the outskirts of the city, and inner deliveries are performed by electricity-powered cargo bikes. Also, the use of unmanned aerial vehicles (drones) can be envisaged: their range is still very limited, such that they might have to fill the batteries en-route to perform all deliveries [Coelho et al., 2017; Wang et al., 2017].

The whole problem seeks to find good integrated solutions for first and second level routes, and the selection of satellites and charging stations.



(a) SDVRP / first level of (E)2E-VRP

(b) MDVRP / second level of (E)2E-VRP

(c) MD/G-VRP / second level of E2EVRP

(d) E2EVRP I

(e) E2EVRP II

(f) E2EVRP III

Figure 1: Subproblems related to the E2EVRP, and different solutions depending on which intermediate facilities or charging stations are used

3. Literature Review

To the best of our knowledge, this problem has not been studied to this date in the literature. We will provide an overview of literature on two-echelon distribution systems as well as electric vehicles and the selection of charging stations en route.

Recently, Cuda et al. [2015] published a survey on different two-echelon structured transportation problems. They cover two-echelon location routing problems (2ELRPs), 2EVRPs, and truck and trailer routing problems (TTRPs) and give a detailed overview of the papers related to the 2EVRP. They provide mathematical formulations and identify the main variants studied in the literature. Based on their findings we will focus on more recent publications.

Zeng et al. [2014] developed a greedy randomised adaptive search procedure with a route-first cluster-second splitting algorithm and a variable neighbourhood descent for the 2EVRP. Their results are good, but unfortunately the algorithm was only tested on the smaller benchmark instances with up to 50 customers.

Breunig et al. [2016] implemented a LNS for 2EVRPs and the two-echelon location routing problem with single depot (2ELRPSD). They used six destroy and one repair operator as well as some well-known local search procedures. We used the basic structure of the algorithm for our heuristic, although the data structure as well as the operators used have been heavily improved and adapted to fit the properties of the proposed problem.

Wang et al. [2017] studied an extension of the 2EVRP, introducing stochastic demands. It is described as a stochastic program with recourse. A genetic algorithm was developed to solve the problem. Their results on the problem with stochastic demands were compared to the best known solutions of the deterministic case, adapting the evaluation of the solution quality.

Recently, the use of vehicles with alternative fuels became more popular. There have been several studies on so called "green" vehicle routing problems (VRPs), focusing on different aspects of environmentally friendly transport. Erdogan and Miller-Hooks [2012] proposed the green vehicle routing problem (GVRP). They extended the VRP by using vehicles with limited range, which have to visit special refuelling stations en route. They also provided insights on the implications of different availability and dispersion of refuelling stations.

Irnich and Desaulniers [2005] deal with shortest path problems with resource constraints (SPPRCs) using a labelling algorithm with dynamic programming. Paths are systematically extended to all feasible directions. The efficiency of such an algorithm heavily depends on its ability to identify and discard paths which are not useful.

Schneider et al. [2013] explicitly deal with typical properties of electric vehicles. Those usually have a limited range and limited freight capacities due to the batteries used to store energy. Vehicles can be recharged during their trips at dedicated charging stations. This process is more time consuming than refuelling a conventional car. In their problem setting, each of the customers has a defined time window when he should be visited. A set of benchmark instances is presented and solutions are generated by using a hybrid heuristic, combining variable neighbourhood search (VNS) with a tabu search (TS) heuristic.

Hiermann et al. [2015] proposed the electric fleet size and mix with fixed costs and time windows (EFSMFTW), where delivering goods to customers can be performed by using a mix of different vehicle types. These differ in acquisition cost, freight capacity, and battery size. The benchmark instances are solved using both a branch-and-price and a hybrid heuristic. The basic components of the latter are adaptive large neighbourhood search (ALNS), with local search and a bi-directional labelling algorithm for intensification.

For further resources on battery electric vehicles, alternative charging modes and other other characteristics and decisions, we refer to the recent survey by Pelletier et al. [2016] and the thesis of Montoya [2016].

4. Solution Method

The proposed metaheuristic is based on LNS [Shaw, 1998], which follows the basic principle of ruin and recreate. Iteratively, a solution will be partially destroyed and repaired. In every iteration, the incumbent solution will be destroyed by a selected destroy operator (Section 4.1), then repaired again with the repair operator (Section 4.2), without considering charging levels. At this point a solution is found, which is feasible in terms of capacities, but possibly infeasible in terms of charging levels. Then a labeling algorithm is applied (Section 4.5) to optimally insert charging stations location in each given route. Afterwards, a local search (Section 4.4) is applied to search for improvements. Algorithm 1 summarizes the structure of the proposed method. We will now describe each methodological component.

Algorithm 1: LNS-E2E

$\mathcal{S}^{best} \leftarrow \mathcal{S} \leftarrow labelling + LS(repair(instance))$ `/* initial solution */`
repeat
    for $i \leftarrow 0$ **to** i_{max} **do**
        $\mathcal{S}^{temp} \leftarrow labelling + LS(repair(destroy(\mathcal{S})))$
        if $cost(\mathcal{S}^{temp}) < cost(\mathcal{S})$ **then**
            $\mathcal{S} \leftarrow \mathcal{S}^{temp}$ `/* accept better solution */`
            $i \leftarrow 0$ `/* reset re-start period */`
    if $cost(\mathcal{S}) < cost(\mathcal{S}^{best})$ **and** $\mathcal{S}$ *is feasible* **then**
        $\mathcal{S}^{best} \leftarrow \mathcal{S}$ `/* store best solution */`
    else
        $\mathcal{S} \leftarrow labelling + LS(repair(instance))$ `/* re-start: new sol. */`
until $time > time_{max}$
return $\mathcal{S}^{best}$

4.1. Destroy operators

The algorithm LNS-E2E makes use of five destroy operators, of which the following three are chosen at each iteration with equal probability:

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. [cf. Breunig et al., 2016]

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 $[0, \lceil p_2 \cdot \sum_{c \in C} \frac{d_c}{Q^2} \rceil]$, where d_c is the demand of customer c and Q^2 the capacity of a second level vehicle. The last term gives a lower bound on the number of routes needed to serve all customers. [cf. Breunig et al., 2016]

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 $\hat{p}_3$ [cf. Breunig et al., 2016].

Additionally, there are two more operators, which can be selected in addition to one of the destroy operators described above:

Open all satellites With a probability of $\hat{p}_4$, all currently closed satellites are opened and available again in the next repair phase.

Remove single customer routes This operator removes all routes which contain only one single customer. Typically, a complete solution will not contain any routes matching this criterion, but it can be the case after partial destruction. So with a probability of $\hat{p}_5$, 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 the short routes allows to use a vehicle originating from another satellite in the next repair phase. [cf. Breunig et al., 2016]

4.2. Repair operator and initial solution

Whenever a solution has to be repaired, we use a simplified version of the well-known Cheapest Insertion Algorithm. The classic cheapest insertion calculates every possible insertion position for every node to insert and selects the move with the lowest increase in cost. The simplified version that we use cuts down complexity: it considers the nodes to be inserted sequentially, and calculates the best possible insertion only for the current node. The output of this construction method depends on the order of nodes to insert —making it also more diverse.

The order of nodes to insert is randomly selected, except in one case. It can happen, that all available vehicles already have itineraries which use almost all of the capacity. The repair operator may not succeed due to a customer with a very high demand and insufficient residual capacities in the routes (we remind that there are no split deliveries allowed on the second level). If this happens, the insertion process is reverted, and all nodes to insert are ordered by descending demand. By inserting customers with high demand first, the construction operator tends to overcome this feasibility issue.

The repair operator does not take charging levels into account. Finally, the creation of the initial solution is performed with the same operators, as it can be assimilated to the repair of a totally destroyed (empty) solution.

4.3. Construction of first level tours

Once the itineraries for the second level have been determined, the quantities needed at the satellites are known. With this information, the first level routes can then be created. Even in the largest benchmark instances, there are only up to ten satellites, so it is relatively easy to find near-optimal solutions for this problem, even with a simple procedure. On the first level, split deliveries are allowed and even often necessary to produce 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. For these cases, we use a 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 a truck's capacity. The same insertion procedure as the repair operator is then 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. This simple technique has also been used in Breunig et al. [2016] and yields good results in very limited CPU time.

4.4. Local search

After both levels have been repaired, we apply a local search procedure based on *2-opt*, *2-opt**, *Relocate*, *Swap* and *Swap2-1*. A first-improvement acceptance policy is used, and the search continues until no improving move can be found. Similarly to the granular search by Toth and Vigo [2003], the moves are restricted to node pairs (i, j) such that j belongs to one of the $\Gamma = 25$ closest vertices from i. During each move evaluation, the method first computes the new distance without changing the current position of the charging stations. If a move leads to a degradation of the solution quality by less than 3%, then the algorithm applies the labeling algorithm (Section 4.5) to find a better position of the charging stations which may have the potential of leading to an improvement.

4.5. Planning of charging stations

At this point the algorithm has found a solution, which is feasible in terms of capacities, but charging levels of city freighters are relaxed. It then uses a dynamic programming labelling algorithm, which finds the optimal positions for charging stations with respect to the given vehicle itinerary. The forward labelling procedure is adapted from Hiermann et al. [2015].

For each pair of second-level nodes, and before routing optimization, the algorithm pre-computes a set of non-dominated charging stations which could be visited between these two nodes. Given a route, forward labelling is performed. Starting from the satellite, labels are extended for visiting the next customer directly, or any of the potential charging stations in between. Each label stores the current cost of travelling along the respective route and the energy usage since the last visit of a charging station, as well as a reference to the previous label. Once all labels have been created at a next node, dominance rules are used to reduce the number of labels.

In rare cases, a given route cannot be made feasible in terms of charging levels. We illustrate a simple example in Figure 2. If the maximum range of a vehicle is 10 units, and no further charging stations are within close proximity, the route on the right hand side can not be made feasible by the labelling algorithm. In Figure 2b the vehicle is short of one unit of distance to reach the next node, in this case the red squared satellite, where a charging facility is also located. Multiplied by a penalty factor p_6, this distance is added to the cost of the route, and the route is flagged as infeasible.

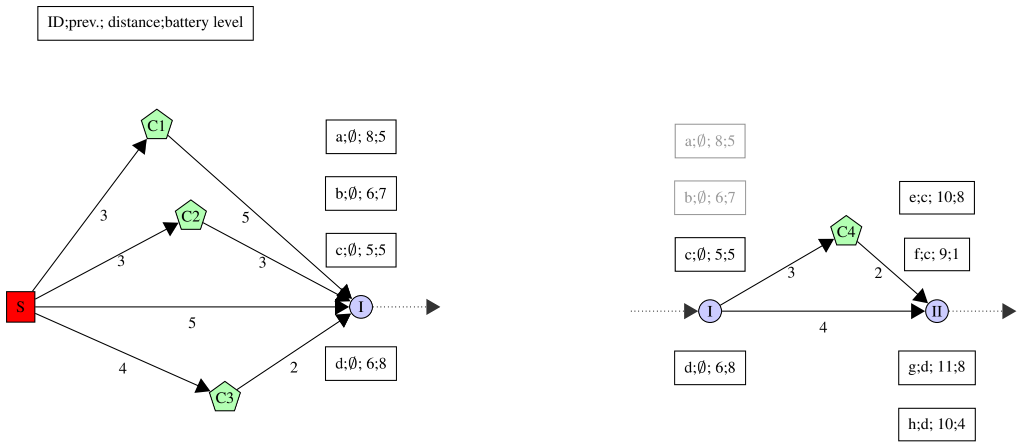
(a) max. d = 10

(b) max. d = 11

Figure 2: A vehicle with a range of $d = 10$ can serve the route of the left example, but not the route on the right.

Figure 3 illustrates the labeling algorithm. The depicted labels show the cost, as well as the current charging level. Again, we assume a battery capacity of 10 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 3 charging stations (green pentagons *C1*-*C3*), shown in Figure 3a. Label *a* at node *I* stores the driving cost of 8 (*S* to *C1*: 3 + *C1* to *I*: 5) and a battery level of 5. Starting at full capacity at *S*, the charging station can be reached at a level of 7; the vehicle then fully charges up to 10 again and after going 5 more units reaches customer *I* at a level of 5. 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 7 and a charging level of 8 (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 6, but at a higher charging level. The remaining labels are then extended to the next node on the route: customer *II* in Figure 3b. 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*.



(a) S to I

(b) I to II

Figure 3: Illustration of the labelling algorithm

5. Benchmark instances

We propose a set of 18 benchmark instances for the E2EVRP. We created those instances based on the 2EVRP instances by Hemmelmayr et al. [2012], also known as "Set 5" instances.

Depot, satellite and customer information remains unchanged. The only additional information relates to the characteristics of the electric city freighters, i.e., their maximum charging level and energy consumption, as well as the coordinates of charging stations. For the sake of simplicity, the energy consumption per unit of distance is normalized to be equal to one unit of energy. The selection and number of locations for charging stations is similar to the way Schneider et al. [2013] created the benchmark instances for the electric vehicle routing problem with time windows and recharging stations (EVRPTWRS), generating approximately one charging station for every ten customers in the instance (10%). Firstly, every depot and satellite location provides charging abilities for city freighters. To decide for the remaining locations, we defined a uniformly distributed grid of 100×100 potential locations based upon the range of x- and y-axis coordinates from the existing 2EVRP instances. For each of those points, we calculated how many customers are in "close proximity" —which was defined by half the average tour length in the best known solution. The more customers one candidate location had in proximity, the more likely it is selected as a charging station. This was achieved by a roulette wheel selection of additional charging stations among those 10,000 potential points.

6. Computational Results

First, we performed some preliminary experiments to produce good values for the new parameters of the algorithm. We used a meta-calibration based on the covariance matrix adaptation evolution strategy (CMA-ES) by Hansen [2006]. It uses the algorithm as a black box solver, feeding different parameter values and receiving the corresponding objective values. In this meta-calibration process the parameters are the decision variables, and the objective function is computed by running the method several times on a set of training instances in each iteration. We used six training instances, two of each of the different size classes (100/200 customers; 5/10 satellites).

After this preliminary calibration, we evaluated the proposed algorithm on the complete set of benchmark instances. These results are displayed in Table 1. The first set of columns present some properties of the instances: number of customers C, satellites S, trucks T, electric city freighters CF and charging stations Ch. Column BKS gives the objective value of the best known solution for the classic 2EVRP without electric vehicles and charging stations. The three solutions marked with an asterisk have been proven to be optimal, and are therefore a lower bound for the solutions with charging stations. The next column "addCh" shows the objective value after the labeling algorithm has been applied to the best known solution for the problem without electric vehicles. This makes the classic solution feasible in terms of charging levels ex post, showing objective values in column *addCh*.

The algorithm uses some pseudo-random numbers, and thus we performed five independent runs on each of the instances with a different initial seed, and show the average objective value in column *avg. 5*, as well as the best solution found within those five runs in column *best 5*. Column *best* shows the current best known solution for each respective instance, which was found during all experiments and parameter tuning. A small difference between the average solution of a heuristic and the best known solutions found over multiple runs is usually a sign of good performance. *T(s)* is the overall wall-clock runtime limit in seconds, and *T*(s)* gives the average time when the best solution was found.

We also report two average gap measures, which aim to illustrate to what extent a combined optimization of route decisions and charging station placements can help to achieve better results. We thus report for each instance the value "Gap avg", calculated as $\frac{avg.5 - addCh}{addCh}$, where "avg. 5" is the average solution value of the new algorithm with joint routing and charging stations optimization, and "addCh" is the solution cost obtained by the classical 2EVRP solution method with a post-optimization of charging stations. Similarly, "Gap best" is calculated as $\frac{best.5 - addCh}{addCh}$, and thus illustrates the percentage gap between the value of our best solution out of five runs and the post-processed BKS solution of the 2EVRP.

Table 1: E2EVRP Results

						Classic 2EVRP		Electric 2EVRP						
Instance	C	S	T	CF	Ch	BKS	addCh	avg. 5	best 5	BKS	T(s)	T*(s)	Gap avg	Gap best
100-5-1	100	5	5	32	10	1564.46*	1672.32	1628.91	1617.08	1613.37	900	540	-2.60%	-3.52%
100-5-1b	100	5	5	15	10	1108.62	1244.26	1213.77	1206.64	1187.22	900	431	-2.45%	-4.58%
100-5-2	100	5	5	32	10	1016.32*	1119.43	1060.44	1058.71	1058.71	900	416	-5.27%	-5.42%
100-5-2b	100	5	5	15	10	782.25	850.62	847.30	847.01	830.33	900	319	-0.39%	-2.39%
100-5-3	100	5	5	30	10	1045.29*	1086.48	1065.78	1065.38	1065.23	900	169	-1.91%	-1.96%
100-5-3b	100	5	5	16	10	828.54	926.18	909.26	906.34	903.56	900	274	-1.83%	-2.44%
100-10-1	100	10	5	35	11	1124.93	1194.26	1156.55	1141.28	1141.20	900	229	-3.16%	-4.44%
100-10-1b	100	10	5	18	11	916.25	1046.19	1032.73	1016.52	1014.57	900	205	-1.29%	-3.02%
100-10-2	100	10	5	33	11	990.58	1128.09	1060.61	1057.65	1053.67	900	474	-5.98%	-6.60%
100-10-2b	100	10	5	18	11	768.61	931.39	884.70	884.70	862.84	900	479	-5.01%	-7.36%
100-10-3	100	10	5	32	11	1043.25	1122.85	1081.90	1074.09	1069.70	900	193	-3.65%	-4.73%
100-10-3b	100	10	5	17	11	850.92	970.42	928.64	919.55	917.85	900	292	-4.31%	-5.42%
200-10-1	200	10	5	62	20	1556.79	1665.70	1643.67	1631.28	1603.37	900	252	-1.32%	-3.74%
200-10-1b	200	10	5	30	20	1187.62	1321.33	1314.27	1284.77	1269.18	900	445	-0.53%	-3.95%
200-10-2	200	10	5	63	20	1365.74	1407.74	1412.84	1396.42	1388.25	900	490	0.36%	-1.38%
200-10-2b	200	10	5	30	20	1002.85	1089.94	1086.07	1054.50	1049.88	900	269	-0.35%	-3.68%
200-10-3	200	10	5	63	20	1787.73	1828.68	1821.43	1814.25	1809.27	900	349	-0.40%	-1.06%
200-10-3b	200	10	5	30	20	1197.90	1265.78	1265.38	1250.27	1251.30	900	523	-0.03%	-1.14%
Avg.						1118.81	1215.09	1189.68	1179.25	1171.64	900	353	-2.09%	-3.58%

Integrated planning of charging stations during the whole optimization process is almost always beneficial over post-optimization. Even though the base solutions, from where post-optimization starts, are very good (the currently best known solutions to the classic 2EVRP from literature), the integrated optimization can outperform these solutions. In only one out of the 18 benchmark instances, the average objective value over five runs is slightly worse than the ex-post optimized best known solution, but in one of these five runs the optimized solution is 1.38% better

(200-10-2). Solutions are typically found after less than six minutes on average. The overall average improvement, using integrated optimization, compared to post-optimization is slightly above 2%. Moreover, for some instances, improvements greater than 5% were achieved, even reaching more than 7% for one instance.

Table 2: Sensitivity analysis and contribution of individual components

Base	no related	no route	no close	no open	no single
1189.68	2.70%	2.44%	1.52%	3.10%	1.61%

Finally, to illustrate the role of the components of the proposed algorithm, Table 2 displays the contribution of each of the different destroy operators to the overall solution quality. The baseline results correspond to the average percentage gap of the complete method over five runs and all 18 instances. The columns give the deviation of this base value when one specific component of the algorithm is deactivated.

From these results, we observe that the most important method component is the destroy operator which makes previously closed satellites available again (*no open*). Even in this configuration, satellites can be open again if a solution is completely destroyed. Without the explicit use of a dedicated operator to re-open satellites, the solutions are 3.1% worse on average. The operator which closes one satellite has a smaller but still significant impact on the overall solution quality, with a deviation of 1.52% from the base value when deactivated. This can be probably explained by the fact that, in most cases, the choice of satellites is quite obvious, and will be automatically achieved by the cheapest insertion repair heuristic. The *no close* operator is essential in all the other cases, where customers are serviced by a satellite further away. These solutions could not be found without specifically optimizing the choice of used satellites. The configuration *No related* allows to measure the deterioration due to the deactivation of the "related" destruction operator, which destroys specific areas around a seed customer. Analogously, *no route* measures the performance deterioration when the operator targeting random routes is deactivated, and column *no single* shows the impact of deactivating the destruction operator for the back-and-forth (single node) tours. All operators contribute significantly to the performance of the method, since the deactivation of each single component leads to an overall drop of method performance.

7. Conclusion

We introduced a new problem variant: the E2EVRP. It is based on the classic 2EVRP, where the second tier involves smaller vehicles by design, which should be suitable for inner city deliveries. Using electric vehicles for the second echelong allows to reduce externalities (e.g., noise and pollution) in the city centers, while the use of conventional trucks in the first echelon allows to cover long distances. The electric vehicles considered in the second echelon can be recharged en-route, if needed.

We proposed a LNS-based algorithm to efficiently solve the problem. The method exploits five destroy operators, a single repair procedure as well as a forward labeling algorithm to plan visits to charging stations. The parameters of the algorithm were calibrated using meta-calibration techniques. Our experiments, on new benchmark instances derived from classical sets for the 2EVRP, demonstrate the benefits of an integrated optimization method, which jointly considers the optimization of routes and the placement of charging stations.

Future works on the topic could focus on improving the available heuristic techniques, on the exact resolution of some of these instances, or on the extension of the problem to a wider range of settings, including, e.g., additional time considerations and constraints.

References

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

Coelho, B. N., Coelho, V. N., Coelho, I. M., Ochi, L. S., K., R. H., Zuidema, D., Lima, M. S., and da Costa, A. R. (2017). A multi-objective green uav routing problem. *Computers & Operations Research*, in press. ISSN 0305-0548.

Cuda, R., Guastaroba, G., and Speranza, M. G. (2015). A Survey on Two-Echelon Routing Problems. *Computers & Operations Research*, 55:185–199.

Erdogan, S. and Miller-Hooks, E. (2012). A Green Vehicle Routing Problem. *Transportation Research Part E: Logistics and Transportation Review*, 48(1):100–114.

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

Hemmelmayr, V. C., Cordeau, J.-F., and Crainic, T. G. (2012). An Adaptive Large Neighborhood Search Heuristic for Two-Echelon Vehicle Routing Problems Arising in City Logistics. *Computers & Operations Research*, 39(12):3215–3228.

Hiermann, G., Puchinger, J., Ropke, S., and Hartl, R. F. (2015). The Electric Fleet Size and Mix Vehicle Routing Problem with Time Windows and Recharging Stations. *European Journal of Operational Research*, 252(3):995–1018.

Irnich, S. and Desaulniers, G. (2005). Shortest path problems with resource constraints. In Desaulniers, G., Desrosiers, J., and Solomon, M. M., editors, *Column generation*, chapter 2, p. 33–65. Springer US, Boston, MA.

Montoya, J.-A. (2016). *Electric Vehicle Routing Problems: models and solution approaches*. PhD thesis, Université d'Angers.

Pelletier, S., Jabali, O., and Laporte, G. (2016). Goods Distribution with Electric Vehicles: Review and Research Perspectives. *Transportation Science*, 50(February):3–22.

Schneider, M., Stenger, A., and Goeke, D. (2013). The Electric Vehicle Routing Problem with Time Windows and Recharging Stations. *Publications of Darmstadt Technical University, Institute for Business Studies (BWL)*.

Shaw, P. (1998). Using Constraint Programming and Local Search Methods to Solve Vehicle Routing Problems. In Maher, M. and Puget, J.-F., editors, *Principles and Practice of Constraint Programming*, volume 1520 of *Lecture Notes in Computer Science*, p. 417–431. Springer Berlin Heidelberg.

Toth, P. and Vigo, D. (2003). The Granular Tabu Search and its Application to the Vehicle-Routing Problem. *INFORMS Journal on Computing*, 15(4):333–346.

Wang, K., Lan, S., and Zhao, Y. (2017). 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*, p. 1–13.

Zeng, Z.-y., Xu, W.-s., Xu, Z.-y., and Shao, W.-h. (2014). A Hybrid GRASP+VND Heuristic for the Two-Echelon Vehicle Routing Problem Arising in City Logistics. *Mathematical Problems in Engineering*, 2014:1–11.