

## Scheduling electric vehicles

M. E. van Kooten Niekerk<sup>1,2</sup> · J. M. van den Akker<sup>1</sup> ·  
J. A. Hoogeveen<sup>1</sup>

Accepted: 22 May 2017 / Published online: 12 June 2017  
© The Author(s) 2017. This article is an open access publication

**Abstract** The vehicle scheduling problem (VSP) is a traditional problem in public transport. One of the main assumptions is that buses can be operated the whole day without any interruption for refueling etc. Recently, new technological innovations have led to the introduction of electric vehicles (EVs). For these new vehicles, we cannot ignore the need of refueling during the day, as the range of an electric bus is severely limited, because of the capacity of the batteries. In this paper, we study the electric VSP (e-VSP), where we use EVs with a limited range. During the day the batteries can be charged; in this paper we assume that a battery cannot be replaced/substituted. We present two models that differ in the level of detail resembling the actual processes. In our first model, we assume a linear charging process, work with a constant price of electricity during the day, and do not take the effect of the depth-of-discharge on the lifetime of the battery into account. Our second model resembles practice much better: we allow any type of charging process, work with the actual electricity prices, and take the depreciation cost of the battery into account. To keep this model tractable, however, we approximate the exact value of the charge by discretizing it. The refined model can be solved to optimality using integer linear programming for instances of small/medium size, and therefore, we describe two other solution methods based on column generation

---

✉ M. E. van Kooten Niekerk  
M.E.vanKootenNiekerk@uu.nl

J. M. van den Akker  
J.M.vandenAkker@uu.nl

J. A. Hoogeveen  
J.A.Hoogeveen@uu.nl

<sup>1</sup> Department of Information and Computing Sciences, Utrecht University, P. O. Box 80.089, 3508 TB Utrecht, The Netherlands

<sup>2</sup> Qbuzz BV, Piet Mondriaanplein 31, 3812 GZ Amersfoort, The Netherlands

that find good, but not necessarily optimal, solutions for large instances. We have tested our algorithms on real-world instances.

## 1 Introduction

In the last years, the trend is that public transport becomes more and more environmentally friendly. European norms for engine exhaust gases have become stricter over time, as we can read on the website <http://ec.europa.eu/environment/air/transport/road.htm> of the European Commission (2016). Therefore, new inventions like hybrid and fully electric powered vehicles are now introduced.

For electric vehicles (EVs), there are a few pilot projects in the Netherlands, the largest of which is in Utrecht, where three electric buses drive line 2. For a large-scale extension of the use of EVs, a lot of problems have to be solved. One of them is vehicle scheduling. This is mainly because currently batteries in EVs do not have enough capacity for a whole day of driving, so the batteries have to be replaced or recharged during the day.

In this paper, we present two models for scheduling EVs in public transport. Aside from the classic constraints of vehicle scheduling, the models allow us to take into account the specific constraints for EVs. For simplicity, we will consider only one depot and one vehicle type. Extending the e-VSP to multi depot and multiple vehicle types is similar to extending the traditional VSP to multi depot and multiple vehicle types.

The two models that we present differ in the level of detail resembling the actual processes. In our first model, we assume a linear charging process, work with a constant price of electricity during the day, and do not take the effect of the depth-of-discharge on the lifetime of the battery into account. Our second model resembles practice much better: we allow any type of charging process, work with the actual electricity prices, and take the depreciation cost of the battery into account. To keep this model tractable, however, we approximate the exact value of the charge by discretizing it.

The remainder of the paper is organized as follows. In Sect. 2 we present an overview of the literature on the e-VSP. Next, we describe the differences between traditional and electric buses in Sect. 3. In Sect. 4, we present two models and a number of solution methods for the e-VSP, which are tested and evaluated in Sect. 5. For our experiments, we use data provided by De Lijn, a public transportation organization in Belgium. Finally, we present our conclusions and directions for future research in Sect. 6.

## 2 Literature overview

In this section, we discuss some of the relevant literature on the subject of electric vehicle scheduling in public transport. As indicated below, we assume that a trip is characterized by means of specific travel (start and end) times and locations (fixed departure and arrival places) and a task indicates what a vehicle drives on one day.

Both Adler (2014) and Li (2014) discuss the e-VSP, but they consider EVs with a replaceable battery. For solving the e-VSP, they use Column Generation, where the master problem is selecting a set of vehicle tasks in order to drive all trips, and where the subproblem is finding a vehicle task in order to improve the master problem. This approach is similar to ours. For solving the subproblem, they model the trips and possible links between them as a graph and look for a shortest path. For this, the authors use a pricing algorithm based on Restricted Shortest Path, which is known to be NP-hard (Garey and Johnson 1979). In Adler (2014), the author also discusses the Concurrent Scheduling heuristic, which solves the e-VSP very fast, but at the expense of a solution which is between 10 and 15% away from the optimal solution.

In Reuer et al. (2015), the authors solve a version of the VSP with a fleet consisting of electric vehicles and traditional vehicles without range restrictions. They model this as a time-spaced network and allow opportunity charging, where the battery is supposed to be fully charged in 10 min. If it takes longer, they assume that the battery can be replaced in 10 min. They use six different ways of flow decomposition and a heuristic inspired by Adler (2014).

A problem similar to the e-VSP is the Electric Vehicle Routing Problem with Time Windows and Recharging Stations (E-VRPTWRS), which is defined and discussed in Bruglieri et al. (2015) and Schneider et al. (2014). When the trips in the e-VSP are modeled as customers with a time window of width zero in which only the exact departure time is allowed, we obtain an instance of the E-VRPTWRS, which could be solved by the heuristics in these papers. In both papers, Variable Neighborhood Search is applied. In Bruglieri et al. (2015), this is combined with Local Branching and in Schneider et al. (2014), it is combined with Tabu Search and Simulated Annealing. The authors show that their methods are faster than other known methods, but the datasets they use are small in comparison to the datasets that are used in public transport. In order to determine the applicability of their heuristics in public transport, testing should also be done because the heuristics are tailored for the E-VRPTWRS, especially because in these articles, the conclusion is that the heuristic performs better when the time windows are larger.

When we focus on a small part of solving the e-VSP using Column Generation, we find that in Huang and Li (2016) the authors discuss a shortest path method that can be used in the subproblem when solving the e-VSP with Column Generation. The authors assume that the e-VSP is modeled in the same way as in our Model 1 (see Sect. 4.1). Instead of an integer linear programming (ILP)-formulation, as we use it, they use Column Generation, where every column represents the set of trips driven by a vehicle on one day. The subproblem of this Column Generation is finding the best vehicle task to add to the master problem, which is equal to finding the shortest path in the graph, where all constraints of EVs should be fulfilled. The authors describe a label-correcting algorithm for determining this shortest path. In our models we also look for the shortest path in the same graph, but we can find the shortest path in linear time, while the label-correcting algorithm has exponential running time.

For the complexity of these vehicle scheduling problems, results can be found in Lenstra and Rinnooy Kan (1981), where it is shown that the traditional VSP can be

solved in  $\mathcal{O}(n^3)$  time and that the Multi Depot VSP is NP-hard. For the e-VSP, the NP-hardness is proven in Sassi and Oulamara (2017).

As far as we know, there is no research which takes the battery depreciation and the non-linearity in energy and charging into account, combined with electric vehicle scheduling.

### 3 Problem description

The goal of the e-VSP is to determine an optimal vehicle schedule given a set of trips and taking all constraints regarding EVs into account. The constraints of the e-VSP start with the traditional constraints from the VSP. Given a set of trips with fixed departure and arrival places and times and travel times between all places, determine a set of vehicle tasks (what a vehicle drives on one day) where:

- Every trip is assigned to exactly one vehicle task.
- Every vehicle task drives a feasible sequence of trips.
- The overall costs are minimal.

For this paper, we only discuss the single depot situation with a single vehicle type. Using more than one depot or vehicle type is a straightforward extension of the model, similar to modeling the regular multi depot vehicle scheduling problem.

For the e-VSP, the main difference that we have to take into account is that an EV has a battery that contains a limited amount of energy that is typically not enough for a whole day of driving. So we have some additional constraints that should be observed:

- At all times, the amount of energy in the battery should be sufficient to drive to the next charging station or the depot.
- At a given set of locations, the battery can be charged. This takes time and must be done when the vehicle is standing still. It may also be possible to exchange the empty battery with a full one. For this paper, we do not allow battery exchange.

For the traditional VSP, the objective function reflects the cost of a solution, which is the sum of the fixed cost per vehicle needed and the variable cost per kilometer or minute for fuel, maintenance and crew. For the e-VSP, we have fixed costs per vehicle needed and variable costs for the energy cost per kilometer, but the objective function for the e-VSP will also contain the cost of battery depreciation, because the battery has a limited lifespan that is typically much shorter than the lifespan of the vehicle.

In the following sections, we discuss the characteristics of the practical situations that are relevant to the e-VSP.

### 3.1 Energy cost

For charging an electric vehicle, we need electricity. This electricity comes from the electricity grid. As described on website <http://mpoweruk.com> by Lawson (2014), the consumption of electricity is not at the same level over a day; usually there is a peak consumption at about the end of the afternoon. The level of the peak determines the capacity needed for the power grid and the power plants, and therefore we see at various electricity companies a Time-of-Use pricing in order to encourage consumers to use electricity outside the peak hours. Because the price of electricity may vary significantly over the day, we want to include this in our model. In our model, we will not assign the cost to the time when the electricity is consumed, but to the time when the electricity that is taken from the grid is put in the battery, because this time determines the electricity cost.

### 3.2 Charging infrastructure

Aside from electricity, we also need facilities to charge the vehicles. Such a charging station has a connection to the electricity grid and has equipment to transfer the electricity to the vehicle, for example a power cable or an induction loop. Charging stations can be built at any location, as long as there is a connection to the electricity grid and enough space where vehicles can charge. The most likely places are depots and terminals of routes.

Every charging station has associated properties and costs:

- Location. The construction cost of a charging station may vary due to ground prices, cooperation of the authorities and availability of a high-power electricity connection in the vicinity.
- Charging capacity (space). For every location, a maximum number of vehicles can be charged simultaneously. This depends on the space available.
- Charging capacity (energy). The capacity of the electricity connection may vary per charging station. With a larger capacity EVs may be charged faster or more EVs can be charged simultaneously. However, this requires a larger cable and will be more expensive.

For our problem formulation, we assume that the charging stations and their properties are known. Vehicle capacity of a charging station and optimization by determining the optimal charging infrastructure is not part of this paper.

### 3.3 Battery properties

An extensive description of battery properties can be found on websites Lawson (2014) and Buchmann (2014). The most important properties will be described in the following sections.

### 3.3.1 Battery capacity

A battery is manufactured for a given capacity. This is the amount of energy that can be stored at standard operating conditions when the battery is new. When the temperature is low or very high, the capacity can be reduced drastically, by tens of percents. This can be prevented by heating or cooling the battery. When this is not done, we have to take the reduced capacity into account by making different schedules for different available capacities. For example: we create a schedule for the summer and one schedule for the winter, when the capacity is 70% of the usual capacity.

### 3.3.2 Battery lifetime

During the lifetime of a battery, the capacity diminishes because of chemical processes that occur inside the battery due to its usage. The lifetime of a battery is usually specified in the Cycle Lifetime, which is the number of times that the battery can be fully discharged until it is considered end-of-life, which is when, measured at room temperature, the capacity of the battery is 80% of its original capacity. The actual lifetime of a battery is not determined by the number of charge/discharge cycles, but by the amount of energy that has been stored in total. Charging and discharging the battery for 10% can be done ten times the number specified as Cycle Lifetime until the battery reaches its end-of-life.

A second important factor is the Depth of Discharge (DoD). Discharging a battery fully will dramatically reduce its lifetime due to chemical processes that occur in the battery. When we have a battery with a Cycle Lifetime of 1000, then discharging this battery for 10% can be done 10,000 times. In practice, the number will be higher. In Fig. 1, a graph is shown for a Li-ion battery showing the number of recharge cycles related to the DoD. For every amount of energy in the battery (state of charge, SoC), we can calculate the cost per energy unit. When we want to use this in our models, we have to know what the SoC was before and after charging and use the average cost per kWh in this range. The next paragraphs show an example of a calculation. For other batteries, the calculation will be similar.

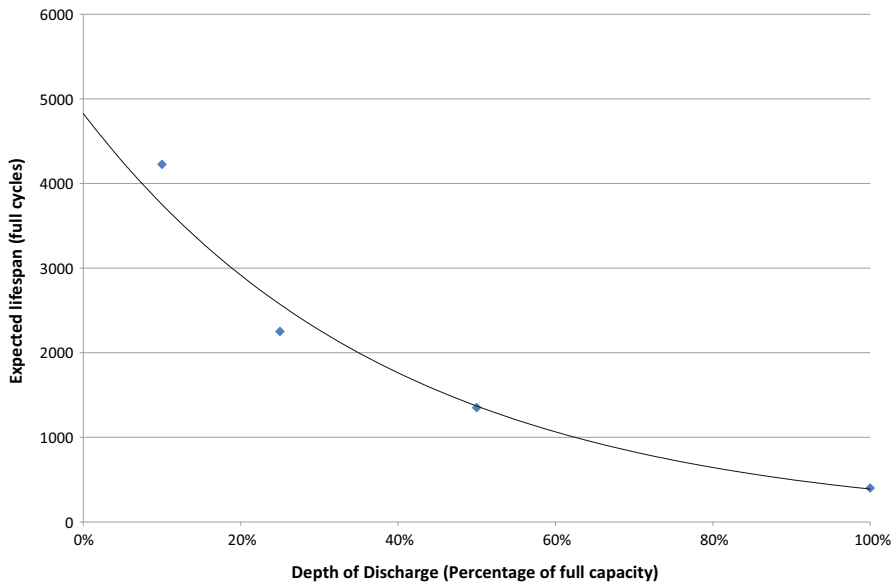
We start with finding a formula that calculates the number of charge/recharge cycles  $cycles(x)$  until end-of-life of the battery given the DoD  $x$  where  $x \in [0, 1]$ . For this, we use the numbers from Fig. 1 and fit a function to it using the minimal-least-squares method. Evaluating miscellaneous function families, we get the best fit for an exponential function. The best fit found for the values shown in the graph is:

$$cycles(x) = 4825.3e^{2.519x}. \quad (1)$$

When we use  $cost_{battery}$  to denote the cost of buying a battery, the cost  $cost(x)$  of one cycle is defined as

$$cost(x) = \frac{cost_{battery}}{cycles(x)} = \frac{e^{2.519x}}{4825.4} cost_{battery}. \quad (2)$$

The cost  $z(x_1, x_2)$  of one cycle of charge/discharge between a DoD of  $x_1$  and a DoD of  $x_2$ , where  $x_1 \geq x_2$  is



**Fig. 1** Graph showing the relation between DoD and the lifetime of a Li-ion battery, measured in number of recharge cycles. This is measured by repeatedly discharging the battery to the DoD on the *horizontal axis*. On the *vertical axis*, the number of charge/recharge cycles until end-of-life is indicated. Graph based on data from page BU808 on Battery University, <http://batteryuniversity.com>

$$z(x_1, x_2) = \text{cost}(x_1) - \text{cost}(x_2) = \frac{e^{2.519x_1} - e^{2.519x_2}}{4825.4} \text{cost}_{\text{battery}}. \quad (3)$$

Note that this formula calculates only the cost related to battery depreciation of a charge/discharge cycle. The cost of the energy itself is not included in this formula.

### 3.3.3 Battery charging characteristics

Charging a battery is not as simple as it may look. For every kind of battery, the chemistry of the battery should be taken into account. A charger for one type of battery is usually not capable of charging another type. For the Li-ion battery (which is the kind of battery that is used for most electric vehicles), a complex charging scheme applies. This charging scheme implies that until 80% of full charge, the battery is charged quickly and after that, charging will be slowed down in order not to overheat the battery. In practice, charging a battery from 0 to 80% is a linear process and will take about the same time as charging it from 80 to 100%, where charging gradually slows down. In Sect. 4.2, we incorporate this in the graph of Model 2 by creating arcs between possible states of charge.

## 4 Models to solve the e-VSP

In order to solve the e-VSP, we model it in two different ways. In Model 1, we use a standard VSP-model, to which we add continuous variables to track the charge of the batteries. Model 1 is described in Sect. 4.1. In Model 2, described in Sect. 4.2, we extend the underlying graph of the VSP-model in order to keep track of the charge. Every node that represents a trip is replaced by a set of nodes where every node represents a possible state of charge on every trip. We use three different approaches to solve Model 2, we call these Models 2a, 2b and 2c. In Model 2a, we formulate the problem as an ILP and solve it with IBM ILOG CPLEX 12.2. In Model 2b, described in Sect. 4.3, we use Column Generation (CG) in order to get a good solution and in Model 2c, we use CG in combination with Lagrangian Relaxation, which also gives a good, but not necessarily optimal solution for the e-VSP.

Each one of these models has different properties. These properties are shown in Table 1.

### 4.1 Model 1: e-VSP with continuous variables for battery charge

In our first model, we model the e-VSP in the same way as the VSP: one node per trip and one node at the garage for every possible arrival or departure time. The nodes at the depot are used to keep track of the number of vehicles parked at the depot. For every combination of trip nodes and depot nodes, we create an arc when these two can be driven by the same vehicle. This arc is a connection between two trips and can contain deadhead trips, charging at a charging station and standing still waiting for the next trip. All costs involved with this are associated with the arc, including the cost of the electricity that is used during the trip and the deadhead trip that may be associated with this arc.

For every trip, we assign an extra variable that keeps track of the charge at the start of a trip. For every node and arc we calculate the difference in charge and use this in the model. Ignoring the battery properties as described in Sect. 3.3, we assume that charging is a linear process and that battery depreciation is linear to the amount of energy used. As we can see in Buchmann (2014), the behavior of the

**Table 1** Properties of e-VSP models

Property	Model 1 e-VSP continuous	Model 2a e-VSP discrete	Model 2b and 2c e-VSP w. CG
Charge variable	Exact	Rounded	Rounded
Time-of-day pricing electricity	No	Yes	Yes
Non-linearity of charging time	No	Yes	Yes
Effects DoD on lifetime	No	Yes	Yes
Maximum problem size	Small/medium	Small/medium	Large
Optimal solution guaranteed	Yes	Yes	No



battery is linear when the battery is charged for 80% or less, so when we use only 80% of the capacity in our model, this assumption is valid. This assumption is necessary because we want to model this as a linear program.

In the graph, we create for every trip  $i$  a node  $n_i$ . The collection of all nodes  $n_i$  is called  $N$ . For every minute  $t \in \{0, 1, \dots, 1919\}$ <sup>1</sup> on the depot we also create a node  $d_t$ . For every two trips  $i$  and  $j$  that may follow each other, we create an arc  $(n_i, n_j)$  with associated variable  $a_{ij}$ . At the depot, for every  $t \in \{0, 1, \dots, 1918\}$  we create arcs  $(d_t, d_{t+1})$  with associated variable  $g_t$  representing the number of vehicles at the depot at time  $t$ . These are single arcs that may represent more than one vehicle:  $g_t$  may be larger than 1. We further use  $g_{1919}$  to represent the total number of required vehicles, because all vehicles will be at the depot at the end of the day. We also create arcs that represent pull-in and pull-out trips from and to the depot. A pull-out arc  $(d_t, n_i)$  is represented by variable  $p_{ti}$ , a pull-in arc  $(n_i, d_t)$  is represented by variable  $q_{it}$ . In our ILP formulation, the variables  $a_{ij}$ ,  $p_{ti}$  and  $q_{it}$  are binary variables and  $g_t$  is an integer variable that indicates the use of an arc.

For every node  $n_i$  we require that there is exactly one incoming and one outgoing arc:

$$\sum_{j:n_j \in N} a_{ji} + \sum_{t=0}^{1919} p_{ti} = 1 \quad \text{for all } i. \tag{4}$$

$$\sum_{j:n_j \in N} a_{ij} + \sum_{t=0}^{1919} q_{it} = 1 \quad \text{for all } i. \tag{5}$$

For every node  $d_t$  with  $t \in \{1, \dots, 1919\}$  at the depot, we make sure that  $g_t$  is equal to the number of vehicles at the depot:

$$\sum_{i:n_i \in N} q_{it} + g_{t-1} = \sum_{j:n_j \in N} p_{tj} + g_t \quad \text{for all } t. \tag{6}$$

Furthermore, we need to add constraints to enforce that for every depot, the number of vehicles at the start of the day is equal to that at the end of the day:

$$\sum_{i:n_i \in N} q_{i1919} + g_{1919} = \sum_{j:n_j \in N} p_{1919j} + g_0. \tag{7}$$

For this case, where we consider only one depot, this equation is obsolete.

Until here, these were the constraints for a standard VSP formulation. For tracking the charge of the battery, we use some additional variables and constraints.

For every trip  $i$ , we have a variable  $x_i \in \mathbb{R}^+$ , indicating the charge at the start, and we have a parameter  $u_i \in \mathbb{R}^+$ , for the usage of energy to drive this trip. For every arc  $a_{ij}$ , we define a parameter  $v_{ij} \in \mathbb{R}^+$  for the usage of energy to drive this deadhead trip and a parameter  $w_{ij} \in \mathbb{R}^+$  for the maximum amount of energy that can be

---

<sup>1</sup> We continue after midnight until the next morning 8 a.m., because many timetables end after midnight. In this way, we are still able to model them.

charged on this arc. For now, we assume that charging takes place after a deadhead. For every arc  $(d_i, n_j)$  we define a parameter  $\pi_{ij} \in \mathbb{R}^+$  and for every arc  $(n_i, d_i)$  we define a parameter  $\rho_{it} \in \mathbb{R}^+$  denoting the energy usage during this arc.

For every trip  $i$ , we require that there is enough energy in the battery to complete the trip:

$$x_i \geq u_i. \tag{8}$$

For every trip  $i$ , the charge at the beginning of the trip may not exceed the maximum charge  $x_{max} \in \mathbb{R}^+$ :

$$x_i \leq x_{max}. \tag{9}$$

For every arc between two trips  $i$  and  $j$ , we make sure that there is enough energy at the start of trip  $i$  in order to drive the deadhead trip:

$$x_i \geq u_i + a_{ij}v_{ij}. \tag{10}$$

Because every trip has an outgoing arc, we may omit Constraint (8). We also calculate the charge at the start of trip  $j$ , where  $M$  is an arbitrary large number. For arcs from trip  $i$  to trip  $j$  we use:

$$x_i - a_{ij}u_i - a_{ij}v_{ij} + a_{ij}w_{ij} + (1 - a_{ij})M \geq x_j. \tag{11}$$

For arcs between the depot and a trip  $j$  and a maximum SoC of  $SoC_{max}$  we use:

$$SoC_{max} - p_{ij}\pi_{ij} \geq x_j. \tag{12}$$

And for arcs between a trip  $i$  and the depot we use:

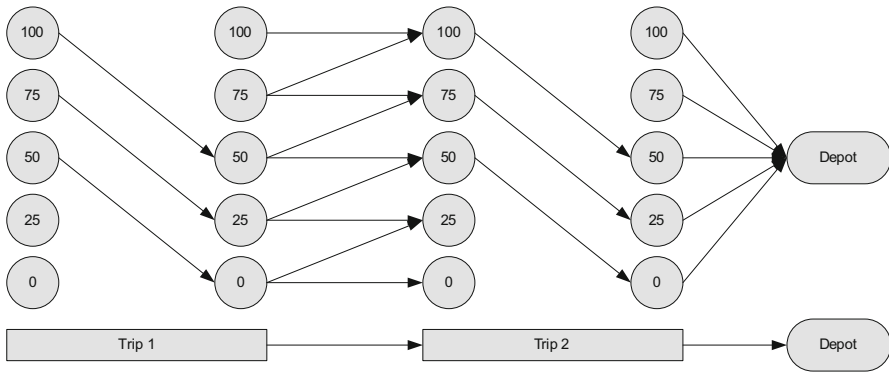
$$x_i - q_{it}u_i - q_{it}\rho_{it} \geq 0. \tag{13}$$

The objective function consists of two terms:

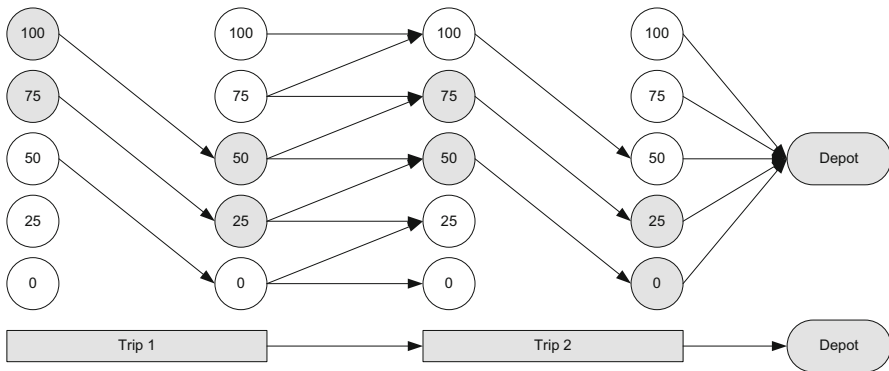
- Fixed costs per vehicle. This is put in the objective by multiplying the costs per vehicle with  $g_{1919}$ , assuming that there are no overnight trips.
- Variable costs per vehicle. This is calculated for every arc and contains the cost of the arc itself and the following trip.

### 4.2 Model 2a: e-VSP with discrete variables for battery charge

The model from Sect. 4.1 does not allow the charging time to be non-linear, does not allow the variable cost to be dependent on the SoC and does not allow Time-of-Day-pricing of energy. Because these factors may have a large impact on the costs, we develop a second model. Our second model is largely similar to Model 1 as described in the first paragraph in Sect. 4.1, but with the difference that we do not keep track of the charge in one single continuous variable per trip. Furthermore, the cost for energy associated with an arc is not the energy used in the trip before and



**Fig. 2** Graph representing two trips and 1 depot at the end. The *circles* are the nodes, representing the start or end of a trip in combination with the charge of the vehicle. Trip 1 and 2 both cost 50% energy and between trip 1 and 2 it is possible to charge either 0 or 25%

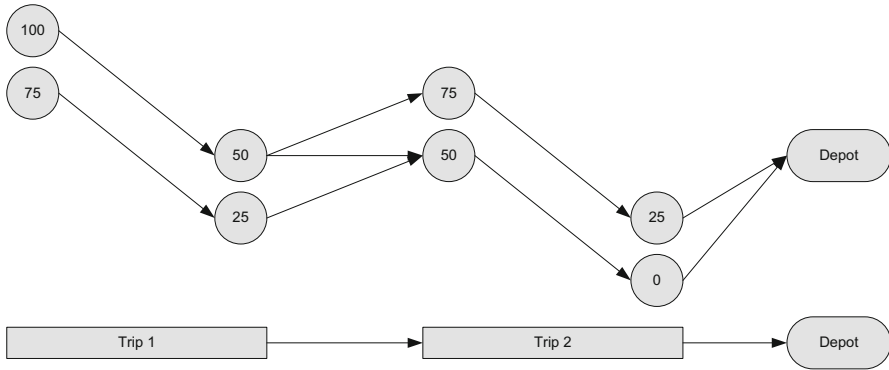


**Fig. 3** The *white* nodes are unreachable, they can be omitted

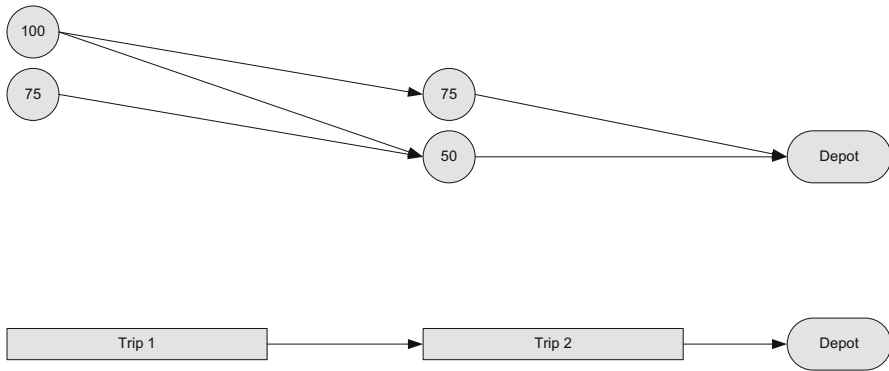
during the arc, but the energy that was charged during the arc. In this way, we can include the Time-of-Day-pricing in our model.

For every trip we create a set of nodes that represent the combination of trip and SoC of the battery at the start of a trip. To keep the size of the model tractable, we discretize the charge for every trip. Therefore, these values are not exact, but now we are able to take most battery properties from Sect. 3.3 into account.

In Figs. 2, 3, 4 and 5, we show the construction and subsequent reduction of the graph used for this problem. We show a graph representing two trips and a depot at the end of the block. In Fig. 2, we have drawn nodes for every start and end of a trip in combination with the electric charge of the vehicle at that moment. Arcs represent the allowed sequence of nodes; if it is possible to charge in between these trips, then there is usually more than one arc per node, because it is possible to charge the vehicle at full power, not charge at all or anything in between. In Fig. 3, we have whitened the nodes that are unreachable, because they do not have any outgoing or incoming arc. In Fig. 4, these nodes have been removed, together with



**Fig. 4** Unreachable nodes and unusable arcs are deleted



**Fig. 5** Trip end nodes are removed, because there is a one-to-one relationship between trip start nodes and trip end nodes

the now obsolete arcs. We note that for every trip, there is a one-to-one relationship between a trip start node and a trip end node, so we combine them to the trip start nodes. This can be seen in Fig. 5.

For every trip  $i$  with a SoC at start of  $x_i \in \mathbb{R}^+$ , we create a node  $n_{ix}$  with associated variable  $\eta_{ix} \in \{0, 1\}$  denoting the use of the node. All these nodes form the collection  $N$ . For the SoC value  $x$ , we only use values that are a multiple of  $\frac{1}{steps}$ , so only  $steps + 1$  different values are possible. The collection of all possible values for the SoC is  $X$ . In Figs. 2, 3, 4 and 5, we use a granularity (referring to *steps*) of 0.25 and in our experiments in Sect. 5 we use a granularity of 0.02. When a combination of trip and charge is not possible, because the vehicle does not have enough charge to complete the trip, the node is not created. For every allowed combination of two nodes  $n_{ix}$  and  $n_{jy}$ , we create an arc  $(n_{ix}, n_{jy})$  with associated variable  $a_{ixjy}$ . In this way, when there is enough time between trips to charge, we can also include the decision whether to charge or not on a charging station, as well as the amount of energy to charge.

For the depot, we also create a set of nodes. For every time unit  $t$  and SoC of  $\sigma \in X$ , we create node  $d_{t\sigma}$ . Between every two adjacent nodes  $d_{t\sigma}$  and  $d_{t+1,\sigma'}$ , where

$\sigma'$  is the SoC when the battery is charged from a SoC of  $\sigma$  during one time unit, we create a depot occupation arc  $(d_{t\sigma}, d_{t+1,\sigma'})$  with associated variable  $g_{t\sigma}$ , which represents the number of vehicles that are at the depot at time unit  $t$  with SoC  $\sigma$ , where  $g_{t\sigma} \in \mathbb{N}^0$ . We also create a depot occupation arc  $(d_{1919\sigma}, d_{0\sigma})$  with variable  $g_{1919\sigma}$ . For deadhead trips from the depot  $d_{t\sigma}$  to an in-service trip  $n_{ix}$  and vice versa we create arcs  $(d_{t\sigma}, n_{ix})$  with variable  $b_{out,t\sigma ix}$  for pull-out trips and  $(n_{ix}, d_{t\sigma})$  with variable  $b_{in,t\sigma ix}$  for pull-in trips. The variables  $\eta_{ix} \in \{0, 1\}$ ,  $a_{ixjy} \in \{0, 1\}$ ,  $b_{out,t\sigma ix} \in \{0, 1\}$  and  $b_{in,t\sigma ix} \in \{0, 1\}$  indicate the use of a node or an arc and are binary.

Using these variables, we formulate the problem as an ILP. Every trip  $i$  should be covered by one vehicle:

$$\sum_{x \in X} \eta_{ix} = 1 \quad \text{for all } i. \tag{14}$$

There should be exactly one arc to every in-service trip  $i$ , if it is used:

$$\sum_{j,y:n_{jy} \in N} a_{jyix} + \sum_{t=0}^{1919} b_{out,t\sigma ix} = \eta_{ix} \quad \text{for all } i, x. \tag{15}$$

And there should be exactly one arc from every in-service trip  $i$ , if it is used:

$$\sum_{j,y:n_{jy} \in N} a_{ixjy} + \sum_{t=0}^{1919} b_{in,t\sigma ix} = \eta_{ix} \quad \text{for all } i, x. \tag{16}$$

For the depot nodes and arcs, the formulas are somewhat different. We assume that nodes  $d_{t\sigma}$  exist for  $t \in \{0, 1, \dots, 1919\}$  and for all  $\sigma$ , being from midnight of that certain day to 8 o'clock in the morning the next day. We require the number of incoming vehicles at every depot node to be equal to the number of outgoing vehicles:

$$\sum_{i,x:n_{ix} \in N} b_{in,t\sigma ix} + g_{t\sigma} = \sum_{i,x:n_{ix} \in N} b_{out,t\sigma' ix} + g_{t+1,\sigma'} \quad \text{for all } t < 1919, \sigma. \tag{17}$$

Then we make sure that the number of vehicles at the start of the day is equal to the number at the end of the day.

$$\sum_{i,x:n_{ix} \in N} b_{in,1919\sigma ix} + g_{1919\sigma} = \sum_{i,x:n_{ix} \in N} b_{out,1919\sigma' ix} + g_{0\sigma'}. \tag{18}$$

### 4.3 Model 2b: Column Generation

Both Models 1 and 2a in Sects. 4.1 and 4.2 give good solutions for the e-VSP. However, in practice we see that both methods are too slow to be applicable in real-life instances with more than 10 vehicles. In this case, we may use the technique of Column Generation to find a good, not necessarily optimal solution (see for example Desaulniers et al. 2005).

For this, we redefine the ILP we use. First we define the Restricted Master Problem (RMP). We look for the optimal set of vehicle tasks while covering all

trips, where a vehicle task is a set of trips that can be driven by one vehicle on a day, taking all constraints with respect to the charge of the EV into account. We use variables  $b_k \in \{0, 1\}$  for every valid vehicle task  $k \in K$ , where  $K$  is the collection of all valid vehicle tasks in the RMP, that indicates if that vehicle task is used in the solution; we use  $c_k$  to denote the cost of vehicle task  $k \in K$ . Furthermore, we use binary parameters  $r_{ki} \in \{0, 1\}$  to indicate if trip  $i$  is part of vehicle task  $k$ .

For the objective function, we use:

$$\text{Minimize: } \sum_{k \in K} c_k b_k. \quad (19)$$

For every trip  $i$ , we require that it is part of at least one vehicle task:

$$\sum_{k \in K} r_{ki} b_k \geq 1 \quad \text{for all } i. \quad (20)$$

$$b_k \in \{0, 1\} \quad \text{for all } k. \quad (21)$$

When we get a result for Constraint 20 that is larger than one, this will translate in multiple vehicles on one trip. In this case, we choose one vehicle to drive the trip and the other vehicle will drive the trip as deadhead trip.

At the start, we fill the RMP with dummy vehicle tasks, one for every trip. These dummy vehicle tasks each contain one trip; when we run the Column Generation, these columns will be discarded soon.

We start with relaxing the integrality constraints: instead of  $b_k \in \{0, 1\}$ , we require  $b_k \geq 0$ . When solving this linear programming (LP) relaxation of this RMP, we get a dual cost for every constraint, that is for every trip. In order to improve the solution of the RMP, we look for extra columns (vehicle tasks) with negative reduced cost. The reduced cost of a column is the cost  $c_k$  of the column minus the sum of the dual cost of all trips included in the vehicle task.

The subproblem is to find vehicle tasks with negative reduced costs. Hereto, we use the graph from Sect. 4.2 and for every arc that ends in a trip, we subtract the dual cost of that trip. Then we look for the path with minimum cost from  $d_0$  to  $d_{1919}$ . If this cost is non-negative, we cannot improve on the solution of the RMP, else we add the column to the RMP and solve it again. We continue until the master problem has been solved to optimality.

At this moment, we have a set of variables  $b_k$ , which indicate which vehicle tasks are part of the optimal solution. However, we do not have a guarantee that the values of these variables are integral. When we have a fractional result, we have different strategies to end up with an integer solution. We do this by reducing the size of the master problem, making heuristic decisions about arcs, nodes and vehicle tasks. The heuristics that we apply are:

- Inspired by Desrosiers et al. (2014), we analyze the result from Column Generation. For the columns that have a strictly positive result in the RMP, we look for identical rows. This means that the trips belonging to these rows always occur together in a vehicle task. When these occur and these trips are consecutive, we merge these two trips and update the graph.

- If there are columns that have a result in the Master Problem larger than 0.95 then the value of these columns is set to 1, these columns are removed from the RMP and the nodes and arcs that belong to these columns are removed from the graph. Then we solve the RMP again with the same columns. This is similar to the Truncated Column Generation approach as described in Section 5 in Pepin et al. (2009).
- If there is any arc that is used by columns or vehicle tasks with a total value for  $b_k$  in the RMP of at least 0.99, we fix this arc and solve the RMP again with the same columns.
- If there is any arc that is not used by any column or vehicle tasks or is used by columns and vehicle tasks with a total value for  $b_k$  in the RMP of less than 0.01, we remove this arc from the graph and solve the RMP again with the same columns.
- If nothing of the above leads to a reduction of the graph and RMP, we fix all columns with a value of 0.7 or more by setting the value to 1. If there are no such columns, we fix the column with the largest value. Then we solve the RMP again with the same columns.

We repeat these steps until the RMP has an integral solution.

#### 4.4 Model 2c: Column Generation in combination with Lagrangian Relaxation

In Sect. 4.3, we used the duals from the LP relaxation for generation of additional columns. In Huisman et al. (2005), an interesting alternative approach to obtain dual values for use in Column Generation is explained.

In Eqs. 19 and 20, we have defined the master problem for Column Generation:

$$\min \sum_{k \in K} c_k b_k \quad (22)$$

$$\text{s.t. } \sum_{k \in K} r_{ki} b_k \geq 1 \quad \text{for all } i \in N, b_k \in \{0, 1\}. \quad (23)$$

We introduce a *Lagrangian Multiplier*  $\lambda_i \in \mathbb{R}$  for every constraint  $i$  and put the constraints in the objective function:

$$\Phi(\lambda) = \min \sum_{k \in K} c_k b_k + \sum_i \lambda_i \left( 1 - \sum_{k \in K} r_{ki} b_k \right) \quad (24)$$

$$\text{s.t. } b_k \in \{0, 1\}. \quad (25)$$

For every non-negative vector of Lagrangian Multipliers  $\lambda$ ,  $\Phi(\lambda)$  gives a lower bound of the solution of the original LP in Eq. 22. Since we can rewrite the objective to  $\sum_{k \in K} b_k (c_k - \sum_{i: n_i \in N} \lambda_i r_{ki}) + \sum_{i: n_i \in N} \lambda_i$ , we observe that the value for  $\Phi(\lambda)$  in Eq. 24 is minimized when we set  $b_k = 1$  when  $c_k - \sum_{i: n_i \in N} \lambda_i r_{ki} < 0$  and  $b_k = 0$  otherwise. We solve the Lagrangian Dual Problem, which is maximizing  $\Phi(\lambda)$ . For this optimization, we use *subgradient optimization*.

The values for  $\lambda$  when  $\Phi(\lambda)$  is maximal can be used as duals for the Column Generation. However, in practice often duplicate columns are generated because already generated columns will get a reduced cost of zero. When zero is the minimal reduced cost possible, we want a column that is not identical to one that is already in the base of the RMP. In order to prevent this, we apply the heuristic from Freling (1997) and Carraresi et al. (1995) that changes the vector  $\lambda$  so that all current columns will have a non-negative reduced cost and the value of the Lagrangian function  $\Phi(\lambda)$  will not decrease. The heuristic is described in Fig. 1.

```

foreach column  $k$  with  $c_k - \sum_{i:n_i \in N} \lambda_i r_{ki} < 0$  do
   $\delta := \frac{c_k - \sum_{i:n_i \in N} \lambda_i r_{ki}}{\sum_{i:n_i \in N} r_{ki}}$  ;
  foreach  $i$  with  $r_{ki}=1$  do
     $\lambda_i := \lambda_i + \delta$ ;
  end
end

```

**Algorithm 1:** Heuristic to modify Lagrangian multipliers

The rest of the solution approach is equal to Sect. 4.3.

## 5 Computational results

To test our algorithms, we use data from the city of Leuven, provided by De Lijn. We will evaluate the cost of exploiting the urban routes 2, 3 and 600/601 with electric vehicles. In the situation where these routes are all driven by buses running on fossil fuel, 27 buses are needed, which drive 543 trips of in total 337:51 h per day. This schedule is shown in Fig. 6. For our optimizations, we use electric vehicles with two different capacities. Every optimization uses one vehicle type, so every optimization is done for every capacity. The characteristics of the electric vehicles are:

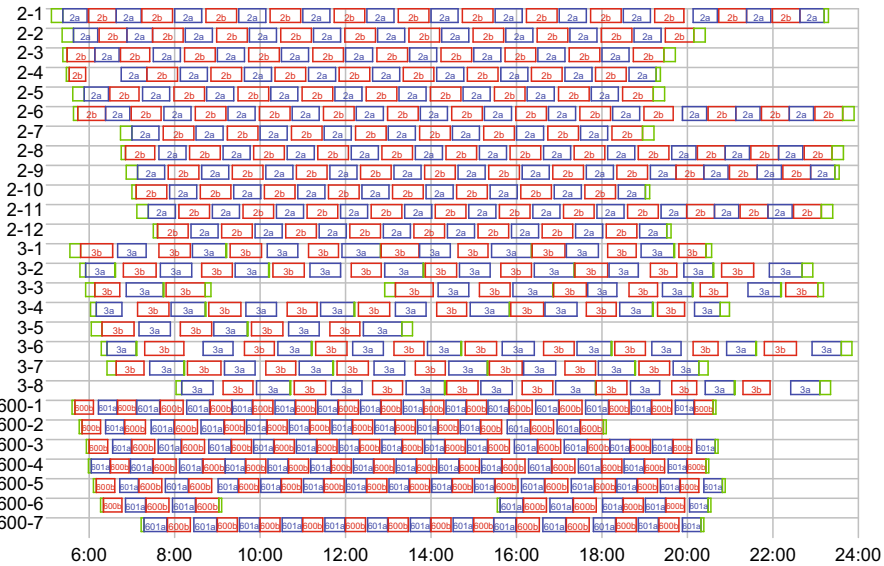
- Battery capacity: 122 or 244 kWh
- Energy usage: 1.2 kWh per km
- Charge speed: 2.0 kWh per minute

For Model 2 we use 51 nodes per trip to reflect the SoC in steps of 0.02. In order to compare Model 1 with these models, we use a linear charging/discharging scheme instead of the more realistic non-linear scheme.

For our calculations, we assume that charging can only take place at four specific sites. When the bus stops at one of those places, the battery will charge. The charging stations are located at

- Kessel-Lo, Hulsberg
- Leuven, Gasthuisberg Campus
- Leuven, Stelplaats Diestsepoort





**Fig. 6** Vehicle schedule for the urban service of Leuven, based on the usage of fossil fuel powered vehicles. On the *horizontal* axis, the time of day is denoted. Every *row* shows the schedule of one vehicle. The *numbers* below the trips denote the route and direction of the trip

– Leuven, Vaartkom

All optimizations are run on a computer with an Intel®Core™i7-3770 microprocessor running on 3.4 GHz, with 16 GB RAM memory. We implemented the algorithms in Java 7, using IBM ILOG CPLEX 12.2 for solving LPs and ILPs.

In order to evaluate our algorithms, we have split the vehicle schedule in three parts, one part with all trips on route 2, one part with the trips on route 3 and one part with all trips on routes 600 and 601. For all three datasets, we executed all four models for both battery capacities. Furthermore, to test the scalability and interlining, we also run Models 3 and 4 on a dataset with all four lines 2, 3, 600 and 601 together. The results of these optimizations can be found in Table 2. For comparison, we have also added the same data for traditional vehicle scheduling in this table. Illustrations of the solutions for traditional vehicle scheduling can be found in Fig. 6 and for Model 4 in Figs. 7 and 8.

When we compare the graphs of the traditional vehicle schedule in Fig. 6 with the one of Model 2c and 244 kWh in Fig. 8, we see that most of the schedule is the same. In the early morning and the early evening, extra layover is added to the schedule in order to charge the vehicle. The capacity of 244 kWh seems to be enough for vehicles that do not drive during the evening, but is not enough for vehicles driving all day. For those vehicles, extra charging time needs to be scheduled.

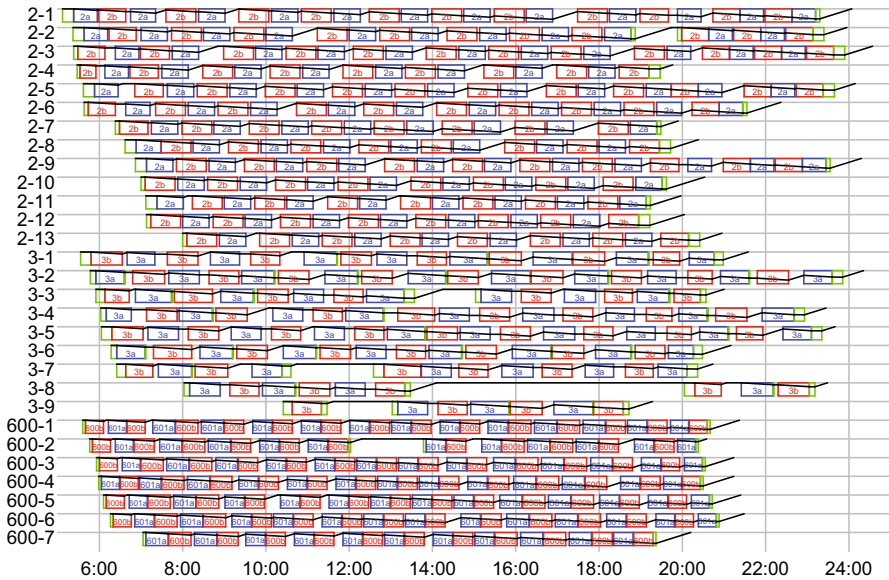
Comparing the traditional vehicles schedule with the one of Model 2c and 122 kWh in Fig. 7 shows that virtually every vehicle will run out of electricity if the

**Table 2** Results for e-VSP

Method	Dataset	Trips	Battery capacity 122 kWh				Total hours	Optim dur. (s)
			Veh.	In-serv hours	Dhd hours	Idle hours		
Trad	Route 2	241	12	159:30	4:32	16:08	180:10	0
Trad	Route 3	127	8	93:58	4:04	16:45	114:47	0
Trad	Route 600/601	175	7	84:23	1:04	7:02	92:29	0
<i>Trad</i>	<i>SUM</i>	543	27	337:51	9:40	39:55	387:26	0
1-MIP	Route 2	241	14	159:30	5:08	26:16	190:54	464
1-MIP	Route 3	127	8	93:58	4:20	21:00	119:18	104
1-MIP	Route 600/601	175	7	84:23	1:04	10:43	96:10	326
<i>1-MIP</i>	<i>SUM</i>	543	29	337:51	10:32	57:59	406:22	894
2a-MIP with discr	Route 2	241	13	159:30	4:44	28:46	193:00	929
2a-MIP with discr	Route 3	127	8	93:58	4:20	23:04	121:22	19
2a-MIP with discr	Route 600/601	175	7	84:23	1:04	12:33	98:00	42
<i>2a-MIP with discr</i>	<i>SUM</i>	543	28	337:51	10:08	64:23	412:22	990
2b-CG	Route 2	241	13	159:30	4:56	28:39	193:05	1045
2b-CG	Route 3	127	8	93:58	4:20	23:48	122:06	141
2b-CG	Route 600/601	175	7	84:23	1:12	13:42	99:17	193
<i>2b-CG</i>	<i>SUM</i>	543	28	337:51	10:28	66:09	414:28	1379
2b-CG	All Routes	543	28	337:51	14:24	64:18	416:33	49330
2c-CG with LR	Route 2	241	13	159:30	4:56	29:21	193:47	102
2c-CG with LR	Route 3	127	9	93:58	5:31	22:52	122:21	27
2c-CG with LR	Route 600/601	175	7	84:23	1:04	13:10	98:37	32
<i>2c-CG with LR</i>	<i>SUM</i>	543	29	337:51	11:31	65:23	414:45	161
2c-CG with LR	All Routes	543	28	337:51	17:42	63:49	419:22	974

Table 2 continued

Method	Dataset	Trips	Battery capacity 244 kWh				Total hours	Optim dur. (s)
			Veh.	In-serv hours	Dhd hours	Idle hours		
Trad	Route 2	241	12	159:30	4:32	16:08	180:10	0
Trad	Route 3	127	8	93:58	4:04	16:45	114:47	0
Trad	Route 600/601	175	7	84:23	1:04	7:02	92:29	0
Trad	SUM	543	27	337:51	9:40	39:55	387:26	0
1-MIP	Route 2	241	12	159:30	4:32	21:19	185:21	4
1-MIP	Route 3	127	8	93:58	4:04	19:45	117:47	0
1-MIP	Route 600/601	175	7	84:23	1:04	9:42	95:09	0
1-MIP	SUM	543	27	337:51	9:40	50:46	398:17	4
2a-MIP with discr	Route 2	241	12	159:30	4:32	26:10	190:12	1325
2a-MIP with discr	Route 3	127	8	93:58	4:04	21:04	119:06	15
2a-MIP with discr	Route 600/601	175	7	84:23	1:04	11:37	97:04	38
2a-MIP with discr	SUM	543	27	337:51	9:40	58:51	406:22	1378
2b-CG	Route 2	241	13	159:30	5:28	20:38	185:36	979
2b-CG	Route 3	127	8	93:58	4:20	18:48	117:06	128
2b-CG	Route 600/601	175	7	84:23	1:04	9:33	95:00	141
2b-CG	SUM	543	28	337:51	10:52	48:59	397:42	1248
2b-CG	All Routes	543	28	337:51	16:54	45:31	400:16	49290
2c-CG with LR	Route 2	241	12	159:30	4:32	19:36	183:38	92
2c-CG with LR	Route 3	127	8	93:58	4:52	18:03	116:53	25
2c-CG with LR	Route 600/601	175	7	84:23	1:04	10:21	95:48	26
2c-CG with LR	SUM	543	27	337:51	10:28	48:00	396:19	143
2c-CG with LR	All Routes	543	27	337:51	18:12	47:43	403:46	539



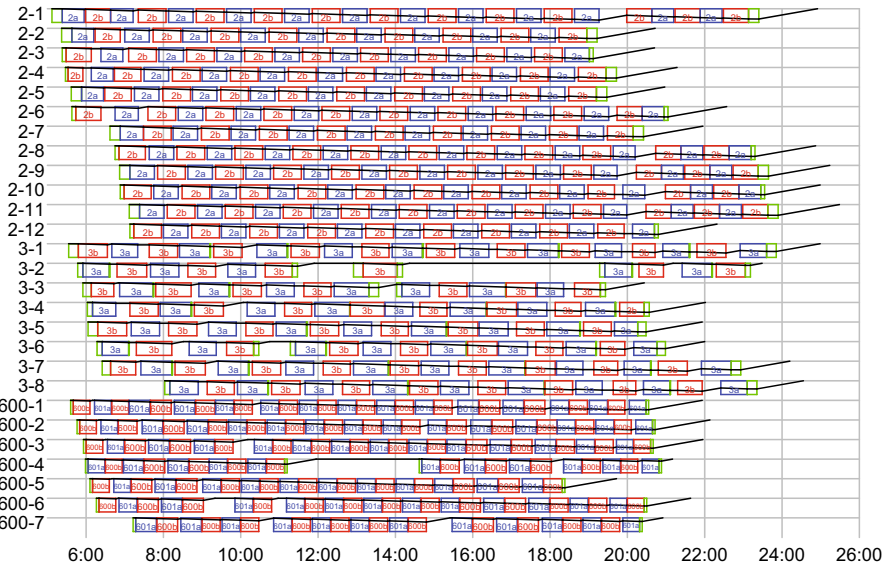
**Fig. 7** Vehicle schedule for the urban service of Leuven, using Model 2c (Column Generation with Lagrangian Relaxation) for optimization and a battery capacity of 122 kWh. On the *horizontal axis*, the time of day is denoted. Every *row* shows the schedule of one vehicle. The *numbers* below the trips denote the route and direction of the trip. The *black line* through the trips show the SoC during the day, where the *top* of the trip is an SoC of 1.0 and the *bottom* of the trip corresponds with an SoC of 0.0

traditional vehicle schedule is used. During the whole day, extra layovers are scheduled in order to charge the batteries.

Comparison of the four models we proposed for solving the e-VSP shows that Model 1 gives the best results, followed by Models 2a, 2c and 2b. It was to be expected that Model 2a gives worse results than Model 1, because Model 2a is an approximation of Model 1. For the complex case of 122 kWh, we see that the run time is comparable. For the easy case of 244 kWh, however, Model 1 is very fast. The disadvantage of Model 1 is that it can not handle non-linear charging schemes; but when this is not necessary, Model 1 is the best choice for instances with an almost sufficient battery capacity. Models 2b and 2c both give comparable results, which are only a few percent more expensive than Models 1 and 2a. The main difference between Models 2b and 2c is the runtime. Model 2c is much faster than Model 2b, and in Model 2c the problem with the full dataset is tractable in reasonable time, while it is not in Model 2b.

## 6 Conclusion

In this article, we have shown that the properties of EVs have to be taken into account, because else we will end up with an infeasible vehicle schedule. For this, we defined the e-VSP in Sect. 3 and see that in most cases, extra vehicles are needed for a feasible vehicle schedule.



**Fig. 8** Vehicle schedule for the urban service of Leuven, using Model 2c (Column Generation with Lagrangian Relaxation) for optimization and a battery capacity of 244 kWh. On the *horizontal axis*, the time of day is denoted. Every *row* shows the schedule of one vehicle. The *numbers* below the trips denote the route and direction of the trip. The *black line* through the trips show the SoC during the day, where the *top* of the trip is an SoC of 1.0 and the *bottom* of the trip corresponds with an SoC of 0.0

We have proposed two models and solution methods to solve the e-VSP and tested them on four datasets. Comparing Model 1 to Model 2a shows that both give us a solution with the same number of EVs, but that the solution of Model 2a needs more waiting hours than Model 1. This waiting time is mostly used for charging the vehicles. Because Model 2a is an approximation of Model 1, it was to be expected that the results were worse than in Model 1. Models 2b and 2c give comparable results, where Model 2c is faster than all the other models. Because Model 2c is capable to optimize the full dataset, since it is faster, and since its final solution is not much more expensive than that of the other models, we advice to use Model 2c for e-VSP. Future work can be done on integrating the properties of charging infrastructure in the model, as described in Sect. 3.2.

**Open Access** This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

## References

Adler JD (2014) Routing and scheduling of electric and alternative-fuel vehicles, Ph.D. thesis. Arizona State University

- Bruglieri M, Pezzella F, Pisacane O, Suraci S (2015) A variable neighborhood search branching for the electric vehicle routing problem with time windows. *Electron Notes Discrete Math* 47(2015):221–228
- Buchmann I (2014) Basic to advanced battery information from battery university. <http://www.batteryuniversity.com>. Accessed 9 Jun 2017
- Carrarese P, Girardi L, Nonato M (1995) Network models, Lagrangean relaxation and subgradients bundle approach in crew scheduling problems. In: Proceedings of the sixth international workshop computer-aided transit scheduling, pp 188–212
- Desaulniers G, Desrosiers J, Solomon MM (2005) Column generation. GERAD 25th Anniversary Series. Springer Science+Business Media, Inc. ISBN 978-0-378-25485-4
- Desrosiers J, Gauthier JB, Lübbecke ME (2014) Row-reduced column generation for degenerate master problems. *Eur J Oper Res* 236:453–460. doi:10.1016/j.ejor.2013.12.016
- Website of the European Commission (2016) European commission—environment—transport and environment. <http://ec.europa.eu/environment/air/transport/road.htm>. Accessed 9 Jun 2017
- Freling R (1997) Models and techniques for integrating vehicle and crew scheduling, Ph.D. thesis. Tinbergen Institute, Erasmus University Rotterdam
- Garey MR, Johnson DS (1979) Computers and intractability: a guide to the theory of NP-completeness. W.H. Freeman, San Francisco
- Huang M, Li J-Q (2016) The shortest path problems in battery-electric vehicle dispatching with battery renewal. *Sustainability* 8(7):607
- Huisman D, Jans R, Peters M, Wagelmans APM (2005) Combining column generation and lagrangean relaxation. In: Desaulniers G, Desrosiers J, Solomon MM (eds) Column generation. GERAD 25th Anniversary Series. Springer, New York, pp 247–270
- Lawson B (2014) Electropaedia, battery and energy technologies. <http://www.mpoweruk.com>. Accessed 9 Jun 2017
- Lenstra JK, Rinnooy Kan AHG (1981) Complexity of vehicle routing and scheduling problems. *Networks* 11(2):221–227. doi:10.1002/net.3230110211
- Li J-Q (2014) Transit bus scheduling with limited energy. *Transp Sci* 48(4):521–539
- Pepin A-S, Desaulniers G, Hertz A, Huisman D (2009) A comparison of five heuristics for the multiple depot vehicle scheduling problem. *J Sched* 12:17–30
- Reuer J, Kliwer N, Wolbeck L (2015) The electric vehicle scheduling problem—a study on time-space network based and heuristic solution approaches. Paper presented at CASPT 2015 conference on advanced systems in public transport, Rotterdam
- Sassi O, Oulamara A (2017) Electric vehicle scheduling and optimal charging problem: complexity, exact and heuristic approaches. *Int J Prod Res* 55(2):519–535. doi:10.1080/00207543.2016.1192695
- Schneider M, Stenger A, Goeke D (2014) The electric vehicle routing problem with time windows and recharging stations. *Transp Sci* 48(4):500–520