Scheduling Electric Vehicles

M.E. van Kooten Niekerk

J.M. van den Akker

J.A. Hoogeveen

Technical Report UU-CS-2015-013
October 2015

Department of Information and Computing Sciences
Utrecht University, Utrecht, The Netherlands
www.cs.uu.nl

ISSN: 0924-3275

Department of Information and Computing Sciences
Utrecht University
P.O. Box 80.089
3508 TB Utrecht
The Netherlands

Scheduling Electric Vehicles

M.E. van Kooten Niekerk J.M. van den Akker J.A. Hoogeveen

October 6, 2015

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. 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 VSP extended by battery constraints. We propose and evaluate several algorithms to solve the VSP for electric vehicles.

1 Introduction

In the last years, the trend is that public transport becomes more and more environmentally friendly. European norms for engine exhaust gases become stricter over time, and new inventions like hybrid and fully electric power supply are now introduced.

For electric vehicles (EVs), in the Netherlands there are a few pilot projects, of which the largest is in Utrecht, where 3 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 a trip.

In this paper, we propose a model for scheduling electric vehicles in public transport. Aside from the classic constraints of vehicle scheduling, the model should 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 remainder of the paper is organized as follows. In Section 2, the differences between traditional and electric buses are described. In Section 3, we propose three models for the Electric Vehicle Scheduling Problem(e-VSP), which are tested and evaluated in Section 4. For our experiments, we use data from De Lijn, a public transportation organization in Belgium. The final conclusion is in Section 5.

2 Problem description

The goal of the e-VSP is to determine the optimal vehicle schedule given a set of trips and taking all constraints regarding EVs into account. The constraints of the e-VSP can be split in several parts. The first part corresponds to the traditional constraints from the VSP:

- Every trip has to be driven by exactly one vehicle. This vehicle should be allowed for this trip.
- Two trips can be driven by one vehicle when they do not overlap and the time between the two trips is large enough. If two consecutive trips planned for one vehicle require a deadhead trip in between, there should be enough time for the deadhead trip.

- When multiple depots are considered, it is required that the number of vehicles of each type at every garage at the start and the end of a day is equal.

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:

- There should be enough energy in the battery.
- 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 costs that are taken into account are 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 cost per vehicle needed and variable cost for the energy cost per kilometer, but we also add 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.

2.1 Energy cost

For charging an electric vehicle, we need electricity. This electricity comes from the electricity grid. As is described on website `http://mpoweruk.com` by Lawson [2014], the consumption of electricity is not equal during the 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 is taken from the grid, because this determines the electricity cost.

2.2 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, co-operation of the authorities and availability of a high-power electricity connection in the vicinity.
- Charging capacity(vehicles). For every location, a maximum number of vehicles can be charged simultaneously. This is dependent 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.

2.3 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.

2.3.1 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.

2.3.2 Battery lifetime

During the lifetime of a battery, the capacity reduces because of chemical processes that occur inside the battery due to the usage of the battery. The lifetime of a battery is usually specified in 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 the original capacity when the battery was new. 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 is 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 10000 times. In practice, the number will be higher. In Figure 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 $c(x)$ until end-of-life of the battery given the DoD x where $x \in [0, 1]$. For this, we use the numbers from Figure 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:

$$c(x) = 4825.3e^{-2.519x} \tag{1}$$

When we use $cost_{battery}$ to denote the cost of buying a battery, the cost $cost(x)$ of one cycle of charge/discharge to a DoD of x is

$$cost(x) = \frac{cost_{battery}}{c(x)} = \frac{e^{-2.519x}}{4825.4} cost_{battery} \tag{2}$$

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

$$z(a, b) = cost(a) - cost(b) = \frac{e^{-2.519a} - e^{-2.519b}}{4825.4} cost_{battery} \tag{3}$$

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.

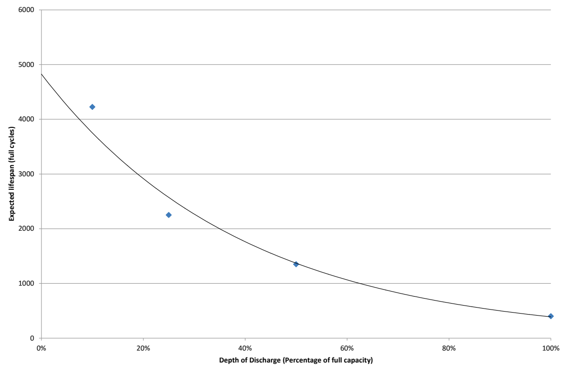

Figure 1: Graph showing the relation between DoD and the lifetime of a Li-ion battery, measured in number of recharge cycles. This is measured by repeatedly discharging the battery to the DoD on the horizontal axis. On the vertical axis, the number of charge/recharge cycles until end-of-life is indicated.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

2.3.3 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 Sections 3.2 and 3.3, we incorporate this in the graphs of models 2 and 3 by creating arcs between possible states of charge.

2.4 Robustness

When compared to the traditional vehicle scheduling problem, the resulting timetable has more requirements on robustness. In traditional vehicle scheduling, possible delays should be considered. But when these occur, one (part of a) trip may be canceled and/or the driver can drive faster. In electric vehicle scheduling, the charge status of the battery is an important factor. If the battery is empty, the vehicle will not drive anymore and has to be towed away. This causes a much larger disruption than a delayed vehicle. Another possible disruption unique to electric vehicles is the inability to charge because of a defective charging station or because of reduced time to charge because of a delay. Another factor is that the energy consumption is also dependent on temperature and on the driving style of the driver.

3 Models to solve the e-VSP

In order to solve the e-VSP, we model it in three 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 Section 3.1. In model 2, we redefine the underlying graph of the VSP-model in order to keep track of the charge. Every discrete possible state of charge on every trip is redefined as a node. We formulate the problem as an ILP and solve it with IBM ILOG CPlex 12.2. Model 2 is described in Section 3.2. In model 3, we use the graph from model 2, but we use Column Generation in order to get a good solution. Model 3 is described in Section 3.3.

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

Property	Model 1 e-VSP continuous	Model 2 e-VSP discrete	Model 3 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

Table 1: Properties of e-VSP models

3.1 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 possible link between two nodes (so from trip to trip or from depot to trip), we create an arc. 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 cost involved with this are associated to the arc, including the cost of the electricity that is used during the trip and the deadhead trip that may be associated to 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. Unlike the battery properties as described in Section 2.3, we assume that charging is a linear process and battery depreciation is linear to the amount of energy used. This assumption is necessary because we want to model this as an LP.

In the graph we use, we create for every trip i a node n_i. For every minute $t \in [0, 1919]$[1] on the depot we also create a node d_t. For every node n_i, we also define a variable c_i that indicates the charge at the start of the trip i. For every two trips i and j that may follow each other, we create an arc a_{ij}. At the depot, for every $t \in [0, 1918]$ we create arcs g_t from garage node t to $t+1$ that represent 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 also 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-in arc from g_t to n_i is represented as p_{ti}, a pull-out arc from n_i to g_t is represented as q_{it}. The variables a_{ij}, p_{ti} and q_{it} are binary variables that indicate the use of an arc.

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

$$\sum_j a_{ji} + \sum_t p_{ti} = 1 \text{ for all } i \tag{4}$$

$$\sum_j a_{ij} + \sum_t q_{it} = 1 \text{ for all } i \tag{5}$$

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

$$\sum_i q_{it} + g_{t-1} = \sum_j p_{tj} + g_t \text{ for all } t \tag{6}$$

[1] We continue after midnight until the next morning 8am, because many timetables end after midnight. In this way, we are still able to model them.

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:

$$\sum_i q_{i1919} + g_{1919} = \sum_j p_{1919j} + g_0 \tag{7}$$

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 c_i, indicating the charge at the start, and we have a parameter u_i, for the usage of energy to drive this trip. For every arc a_{ij} , we define a parameter v_{ij} for the usage of energy to drive this deadhead trip and a parameter w_{ij} for the maximum amount of energy that can be charged on this arc. For now, we assume that charging takes place after a deadhead.

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

$$c_i \geq u_i \tag{8}$$

For every trip i, the charge at the begin of the trip may not exceed the maximum charge c_{max}:

$$c_i \leq c_{max} \tag{9}$$

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

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

Because every trip has an outgoing arc, we may omit Equation (8). We also calculate the charge at the start of trip j, where M is an arbitrary large number:

$$c_i - a_{ij} u_i - a_{ij} v_{ij} + a_{ij} w_{ij} + (1 - a_{ij}) M \geq c_j \tag{11}$$

The objective function consists of three factors:

- Fixed cost per vehicle. This is put in the objective by multiplying the cost per vehicle with g_{1919}, assuming that there are no overnight trips.

- Variable cost per vehicle, excluding energy. This is calculated for every arc and contains the cost of the arc itself and the following trip.

3.2 Model 2: e-VSP with discrete variables for battery charge

The first model from Section 3.1 does not allow the charging time to be non-linear and does not allow the variable cost to be dependent on the SoC. Because these factors may have a large impact on the cost, we develop a second model. Our second model is largely similar to model 1 as described in the first paragraph in Section 3.1, 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 to an arc is not the energy used in the trip before and during the arc, but the energy that us 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 charge at the start of a trip. Because we have to round the charge for every trip, these values are not exact, but on the other hand we are able to take most battery properties from Section 2.3 into account.

In Figures 2 to 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 Figure 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, between trip 1 and trip 2 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 Figure 3, we have whitened the nodes

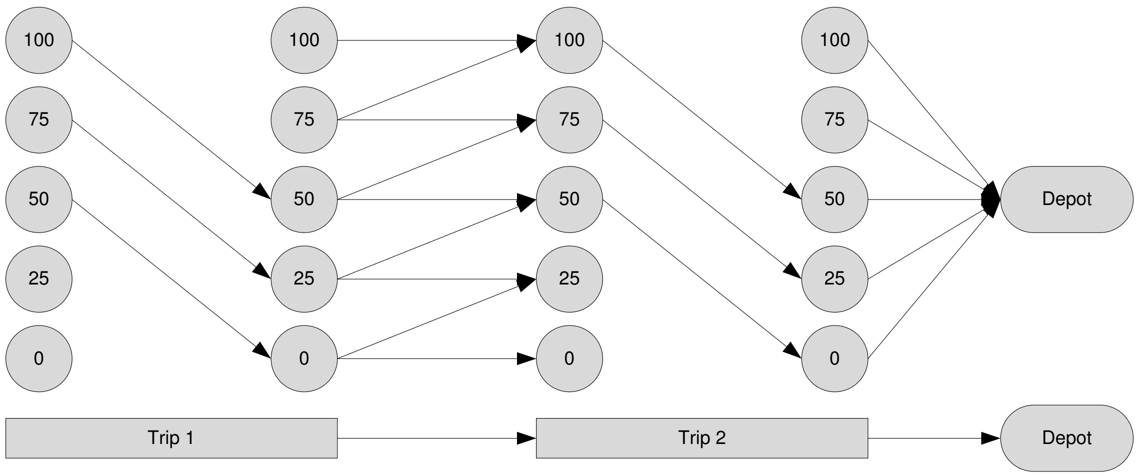

Figure 2: Graph representing two trips and 1 depot at the end. The circles are the nodes, representing the start or end of a trip in combination with the charge of the vehicle. Trip 1 and 2 both cost 50% energy and between trip 1 and 2 it is possible to charge either 0% or 25%.

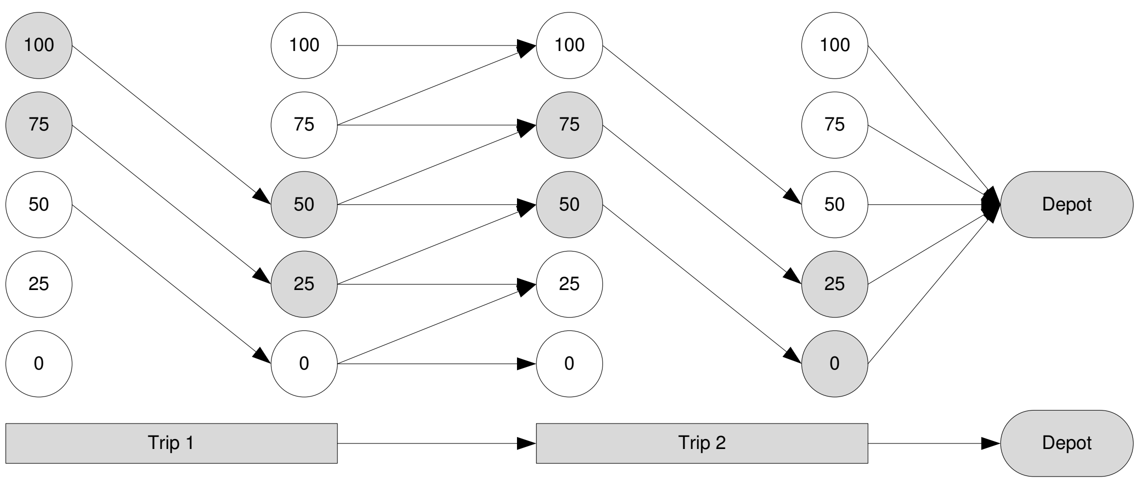

Figure 3: The white nodes are unreachable, they can be omitted.

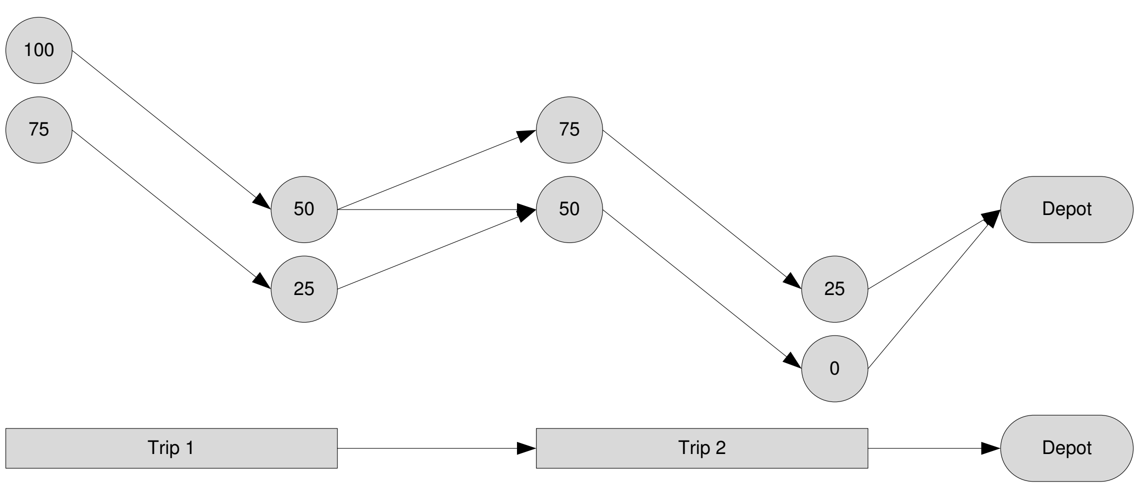

Figure 4: Unreachable nodes and unusable arcs are deleted.

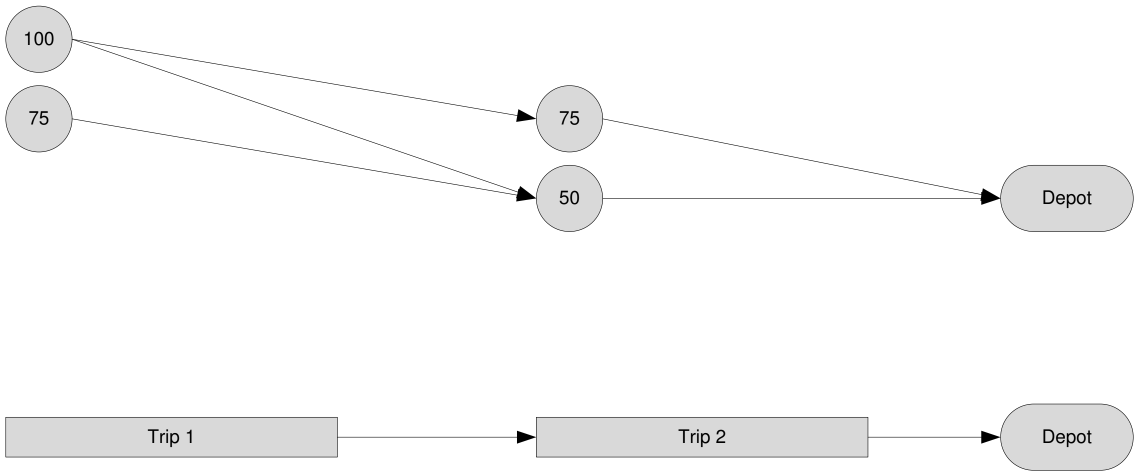

Figure 5: Trip end nodes are removed, because there is a one-to-one relationship between trip start nodes and trip end nodes.

that are unreachable, because they do not have any outgoing or incoming arc. In Figure 4, these nodes are removed, in combination with 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 Figure 5.

For every trip i with a charge at start of j percent, we create a node n_{ij}. For the percentage, we use a granularity of x percent. In Figures 2 to 5, we use a granularity of 25% and in our experiments in Section 4 we use a granularity of 2%. 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_{ij} and n_{kl}, we create an arc a_{ijkl}. In this way, 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, we create node m_t. Between every two adjacent nodes m_t and m_{t+1} we create a depot occupation arc g_t, which represents the number of vehicles that is at the depot at time unit t. Between m_{1919} and m_0 we create a depot occupation arc g_{1919}. For deadhead trips from the depot m_t to an in-service trip n_{ij} and vice versa we create arcs $b_{out,tij}$ for pull-out trips and $b_{in,tij}$ for pull-in trips. For variables m_t and g_t we require them to be integer, the variables n_{ij}, a_{ijkl}, $b_{out,tij}$ and $b_{in,tij}$ 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:

$$\sum_j n_{ij} = 1 \text{ for all } i \tag{12}$$

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

$$\sum_{k,l} a_{klij} + \sum_t b_{out,tij} = n_{ij} \text{ for all } i, j \tag{13}$$

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

$$\sum_{k,l} a_{ijkl} + \sum_t b_{in,tij} = n_{ij} \text{ for all } i, j \tag{14}$$

For the depot nodes and arcs, the formulas are somewhat different. We assume that nodes m_t exist for $t \in \{0, 1920\}$, being from midnight today 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:

$$\sum_{i,j} b_{in,tij} + g_t = \sum_{i,j} b_{out,tij} + g_{t+1} \text{ for all } t < 1919 \ . \tag{15}$$

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.

$$\sum_{i,j} b_{in,1919ij} + g_{1919} = \sum_{i,j} b_{out,1919ij} + g_0 \tag{16}$$

3.3 Model 3: Column Generation

Both Models 1 and 2 in Sections 3.1 and 3.2 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. Column Generation is described in Desaulniers et al. [2005].

For this, we redefine the ILP we use. First we define the Master Problem(MP). 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 $v_i \in \{0, 1\}$ for every valid vehicle task i with cost c_i, that indicates if that vehicle task is used in the solution. We use binary parameters p_{ij} to indicate if trip j is part of vehicle task i.

For the objective function, we use:

$$\text{Minimize: } \sum_i c_i v_i \tag{17}$$

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

$$\sum_i p_{ij} v_i \geq 1 \text{ for all } j \tag{18}$$

$$v_i \in \{0, 1\} \text{ for all } i \tag{19}$$

When we get a result for Equation 17 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.

When solving the LP relaxation of this MP, we get a dual cost for every constraint, that is for every trip. In order to improve the solution of the MP, we look for extra columns (vehicle tasks) with negative reduced cost. The reduced cost of a column is the cost c_i of the column minus the sum of the dual cost of all trips used in the vehicle task.

The subproblem is to find vehicle tasks with negative reduced costs. Hereto, we use the graph from Section 3.2 and for every arc that ends in a trip, we subtract the dual cost of that trip. Then we look for the path with the lowest cost from m_0 to m_{1920}. If this lowest cost is non-negative, we cannot improve the solution of the master problem, else we add the column to the master problem and solve it again. We continue until the master problem has been solved to optimality.

At this moment, we have a set of variables v_i, 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 integer. 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 Master Problem, we look for duplicate 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 remove the duplicates and update the graph by removing arcs that prohibit the involved trips to be in the same vehicle task.

- If there are columns that have a result in the Master Problem larger than 0.95 then these are removed from the MP, and the nodes and arcs that belong to these columns are also removed from the graph. The value of the column is set to 1. Then we solve the MP 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 that have a total value in the MP of at least 0.99, we fix this arc and solve the MP 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 that have a total value in the MP of less than 0.01, we remove these arcs from the graph and solve the MP again with the same columns.
- If nothing of the above leads to a reduction of the graph and MP, 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 MP again with the same columns.

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

3.4 Model 3b: Column Generation in combination with Langrangean Relaxation

In Section 3.3, 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 Equations 17 and 18, we have defined the master problem for Column Generation:

$$\min \sum_i c_i v_i \tag{20}$$

$$\text{s.t.} \sum_i p_{ij} v_i \geq 1 \text{ for all } j \tag{21}$$

$$v_i \in \{0,1\} \tag{22}$$

We introduce a *Lagrangean Multiplier* λ_j for every constraint j and put the constraints in the objective function:

$$\Phi(\lambda) = \min \quad \sum_i c_i v_i + \sum_j \lambda_j (1 - \sum_i p_{ij} v_i) \tag{23}$$

$$\text{s.t.} \quad v_i \in \{0,1\} \tag{24}$$

For every nonnegative vector of Lagrangean Multipliers λ, $\Phi(\lambda)$ gives a lower bound of the solution of the original LP in Equation 20. Since we can rewrite the objective to $\sum_i v_i(c_i - \sum_j \lambda_j p_{ij}) + \sum_j \lambda_j$, we observe that the value for $\Phi(\lambda)$ in Equation 23 is minimized when we set $v_i = 1$ when $c_i - \sum_j \lambda_j p_{ij} < 0$ and $v_i = 0$ otherwise. We will look for λ which gives the maximum value for $\Phi(\lambda)$. For this optimization, *subgradient optimization* is mostly used.

The values for λ can be used as duals for the Column Generation. However, in practice often duplicate columns are generated. In order to prevent this, we apply the heuristic from Freling [1997] and Carraresi et al. [1995] that changes the vector λ so that all current columns will have a non-negative reduced cost and the value of the Lagrangean function $\Phi(\lambda)$ will not decrease. The heuristic is described in Figure 1.

foreach *column i with* $c_i - \sum_j \lambda_j p_{ij} < 0$ **do**
  $\delta := \frac{c_i - \sum_j \lambda_j p_{ij}}{\sum_j p_{ij}}$;
  foreach *j with* p_{ij}*=1* **do**
    $\lambda_j := \lambda_j + \delta$;
  end
end

Algorithm 1: Heuristic to modify Lagrangean multipliers

The rest of the approach of the solution is equal to Section 3.3.

4 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 in total 387 hours per day. This schedule is shown in Figure 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:

- 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 2%. In order to compare Model 1 with Model 2, 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
- 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. The vehicle tasks with numbers 1xx are from route 2, the vehicle tasks with numbers 2xx are from route 3 and those with numbers 6xx are from routes 600 and 601. For all three datasets, we executed all three models for both battery capacities. The results of these optimizations can be found in Tables 4, 6 and 8. For comparison, we have put the same data for traditional vehicle scheduling in Table 2. Illustrations of the solutions for Model 1 can be found in Figures 7 and 8, for Model 2 in Figures 9 and 10 and for Model 3 in Figures 11 and 12.

When we compare the graphs of the traditional vehicle schedule in Figure 6 with the one of model 1 and 244 KWh in Figure 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. We also see that route 600/601 is not changed at all. 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.

Dataset	**Lijn 2**	**Lijn 3**	**Lijn 600/601**	**TOTAL**
Vehicles	12	8	7	**27**
In-service hours	159:30	93:58	84:23	**337:51**
Deadhead hours	4:32	4:04	1:04	**9:40**
Waiting hours	16:08	16:45	7:02	**39:55**
TOTAL hours	**180:10**	**114:47**	**92:29**	**387:26**

Table 2: Results traditional Vehicle Scheduling

Dataset	**Lijn 2**	**Lijn 3**	**Lijn 600/601**	**TOTAL**
Vehicles	14	8	7	**29**
In-service hours	159:30	93:58	84:23	**337:51**
Deadhead hours	5:08	4:20	1:04	**10:32**
Waiting hours	26:16	21:00	10:43	**57:59**
TOTAL hours	**190:54**	**119:18**	**96:10**	**406:22**
Run time optimization	464,1	104,3	326	894,4

Table 3: Results Electrical Vehicle Scheduling, using Model 1 and 122 KWh

Dataset	**Lijn 2**	**Lijn 3**	**Lijn 600/601**	**TOTAL**
Vehicles	12	8	7	**27**
In-service hours	159:30	93:58	84:23	**337:51**
Deadhead hours	4:32	4:04	1:04	**9:40**
Waiting hours	21:19	19:45	9:42	**50:46**
TOTAL hours	**185:21**	**117:47**	**95:09**	**398:17**
Run time optimization	4,1	0,1	0,1	4,3

Table 4: Results Electrical Vehicle Scheduling, using Model 1 and 244 KWh

Dataset	**Lijn 2**	**Lijn 3**	**Lijn 600/601**	**TOTAL**
Vehicles	13	8	7	**28**
In-service hours	159:30	93:58	84:23	**337:51**
Deadhead hours	4:44	4:20	1:04	**10:08**
Waiting hours	28:46	23:04	12:33	**64:23**
TOTAL hours	**193:00**	**121:22**	**98:00**	**412:22**
Run time optimization	928,8	18,8	42	989,6

Table 5: Results Electrical Vehicle Scheduling, using Model 2 and 122 KWh

Dataset	**Lijn 2**	**Lijn 3**	**Lijn 600/601**	**TOTAL**
Vehicles	12	8	7	**27**
In-service hours	159:30	93:58	84:23	**337:51**
Deadhead hours	4:32	4:04	1:04	**9:40**
Waiting hours	26:10	21:04	11:37	**58:51**
TOTAL hours	**190:12**	**119:06**	**97:04**	**406:22**
Run time optimization	1325	15	38	1378

Table 6: Results Electrical Vehicle Scheduling, using Model 2 and 244 KWh

Dataset	Lijn 2	Lijn 3	Lijn 600/601	TOTAL
Vehicles	14	9	9	**31***
In-service hours	160:56	94:42	85:22	**341:00**
Deadhead hours	6:24	4:30	1:52	**12:46**
Waiting hours	33:51	25:32	13:18	**72:41**
TOTAL hours	**201:11**	**124:44**	**100:32**	**426:27**
Run time optimization	251	51,9	98,7	401,6

Table 7: Results Electrical Vehicle Scheduling, using Model 3(Column Generation) and 122 KWh. *Note that the total number of vehicles is one less than the sum of the parts. This is because the results from the separate datasets can be combined in order to save one vehicle

Dataset	Lijn 2	Lijn 3	Lijn 600/601	TOTAL
Vehicles	13	8	8	**29**
In-service hours	159:30	93:58	84:23	**337:51**
Deadhead hours	5:40	5:07	1:20	**12:07**
Waiting hours	27:10	23:45	12:54	**63:49**
TOTAL hours	**192:20**	**122:50**	**98:37**	**413:47**
Run time optimization	201	62	71	334

Table 8: Results Electrical Vehicle Scheduling, using Model 3(Column Generation) and 244 KWh

Dataset	Lijn 2	Lijn 3	Lijn 600/601	TOTAL
Vehicles	14	9	8	**31**
In-service hours	159:30	93:58	84:23	**337:51**
Deadhead hours	5:20	5:02	1:04	**11:26**
Waiting hours	33:41	24:55	14:47	**73:23**
TOTAL hours	**198:31**	**123:55**	**100:14**	**422:40**
Run time optimization	187	40	68	295

Table 9: Results Electrical Vehicle Scheduling, using Model 3b(Column Generation with Langrangean Relaxation and 122 KWh.

Dataset	Lijn 2	Lijn 3	Lijn 600/601	TOTAL
Vehicles	13	8	7	**28**
In-service hours	159:30	93:58	84:23	**337:51**
Deadhead hours	4:44	5:08	1:04	**10:56**
Waiting hours	22:58	21:27	10:34	**54:59**
TOTAL hours	**187:12**	**120:33**	**96:01**	**403:46**
Run time optimization	140	50	46	236

Table 10: Results Electrical Vehicle Scheduling, using Model 3b(Column Generation with Langrangean Relaxation and 244 KWh

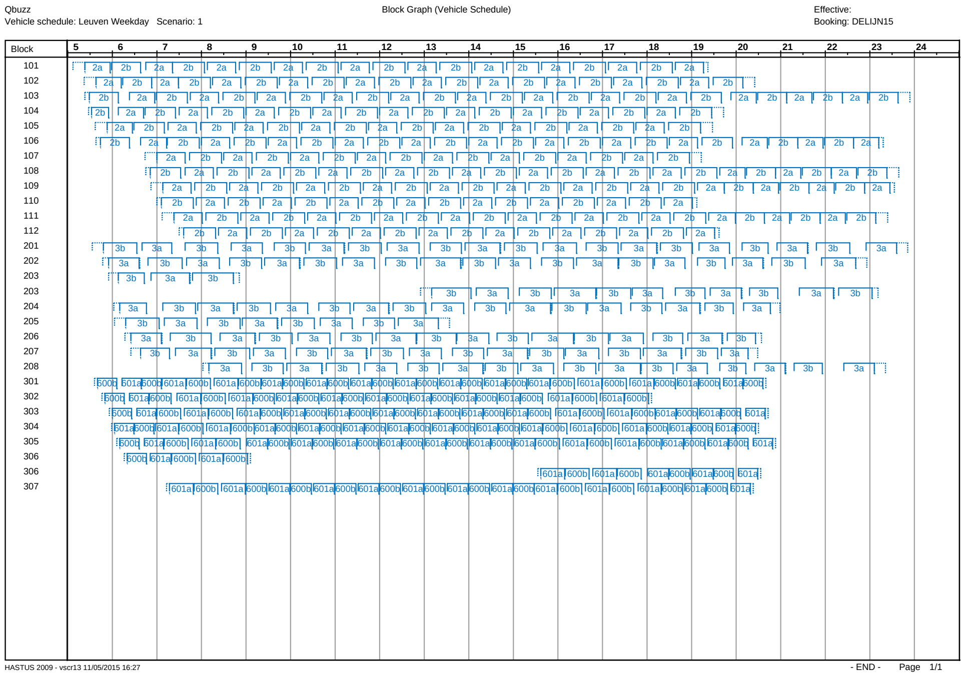

Figure 6: Vehicle schedule for the urban service of Leuven, based on the usage of fossil fuel powered vehicles. On the horizontal axis, the time of day is denoted. Every row shows the schedule of one vehicle. The numbers below the trips denote the route and direction of the trip.

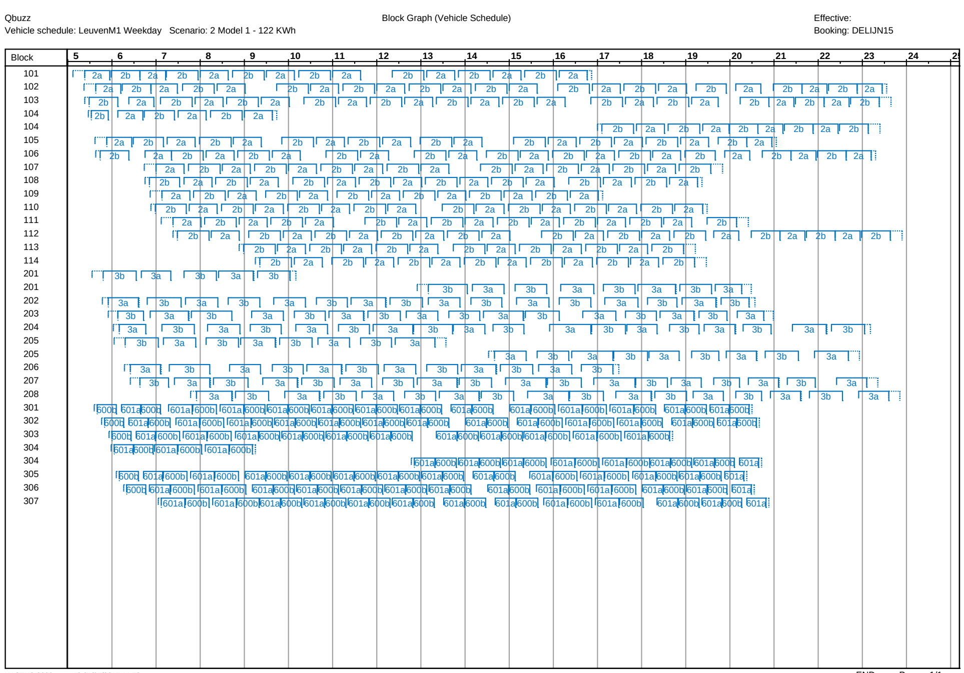

Figure 7: Vehicle schedule for the urban service of Leuven, using Model 1 for optimization and a battery capacity of 122 KWh. On the horizontal axis, the time of day is denoted. Every row shows the schedule of one vehicle. The numbers below the trips denote the route and direction of the trip.

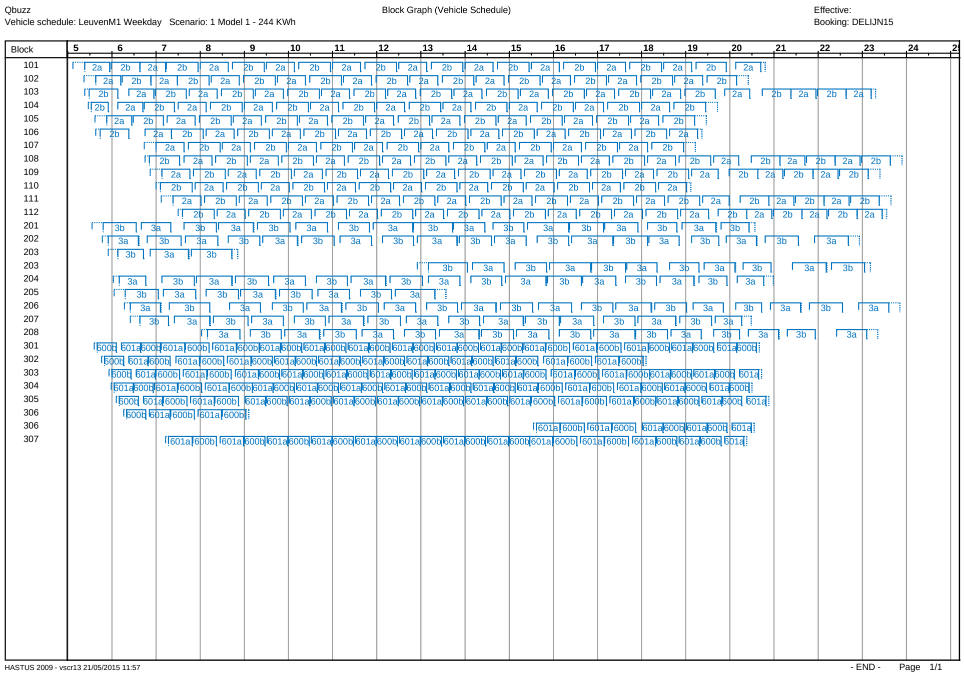

Figure 8: Vehicle schedule for the urban service of Leuven, using Model 1 for optimization and a battery capacity of 244 KWh. On the horizontal axis, the time of day is denoted. Every row shows the schedule of one vehicle. The numbers below the trips denote the route and direction of the trip.

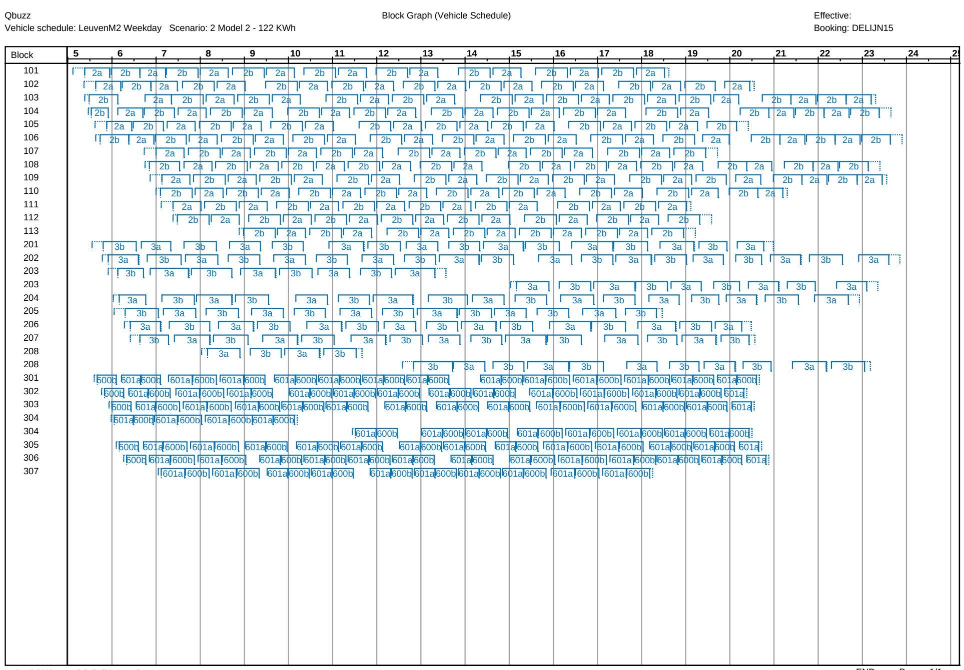

Figure 9: Vehicle schedule for the urban service of Leuven, using Model 2 for optimization and a battery capacity of 122 KWh. On the horizontal axis, the time of day is denoted. Every row shows the schedule of one vehicle. The numbers below the trips denote the route and direction of the trip.

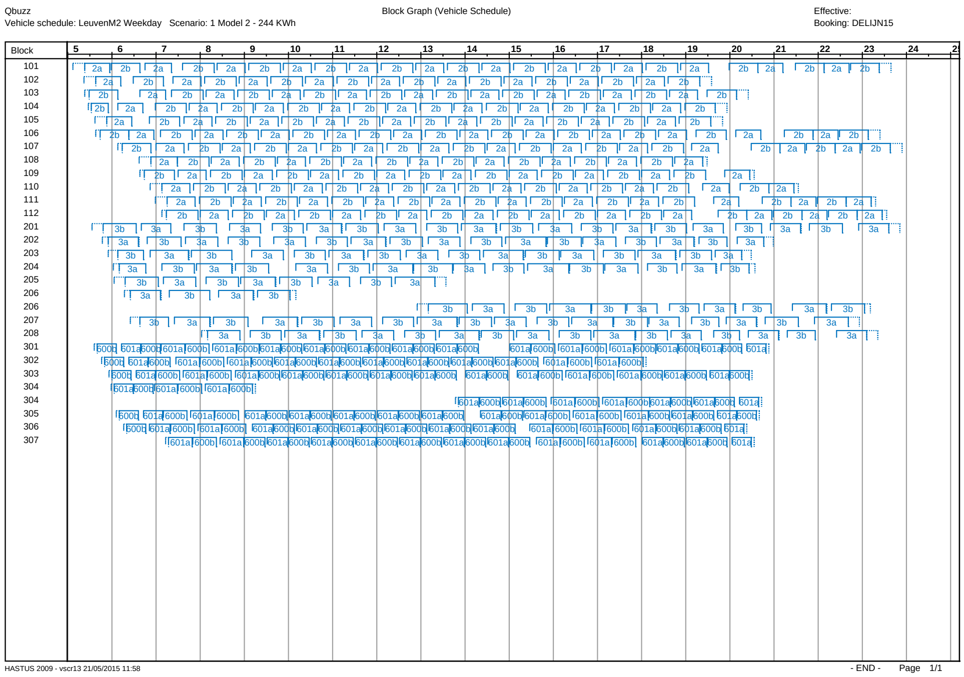

Figure 10: Vehicle schedule for the urban service of Leuven, using Model 2 for optimization and a battery capacity of 244 KWh. On the horizontal axis, the time of day is denoted. Every row shows the schedule of one vehicle. The numbers below the trips denote the route and direction of the trip.

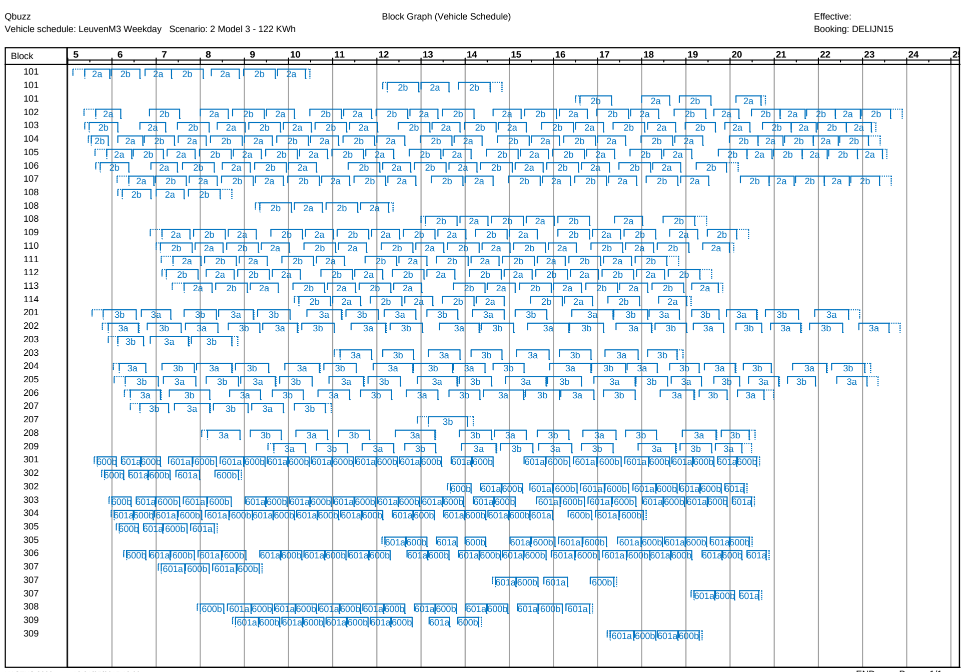

Figure 11: Vehicle schedule for the urban service of Leuven, using Model 3(Column Generation) for optimization and a battery capacity of 122 KWh. On the horizontal axis, the time of day is denoted. Every row shows the schedule of one vehicle. The numbers below the trips denote the route and direction of the trip.

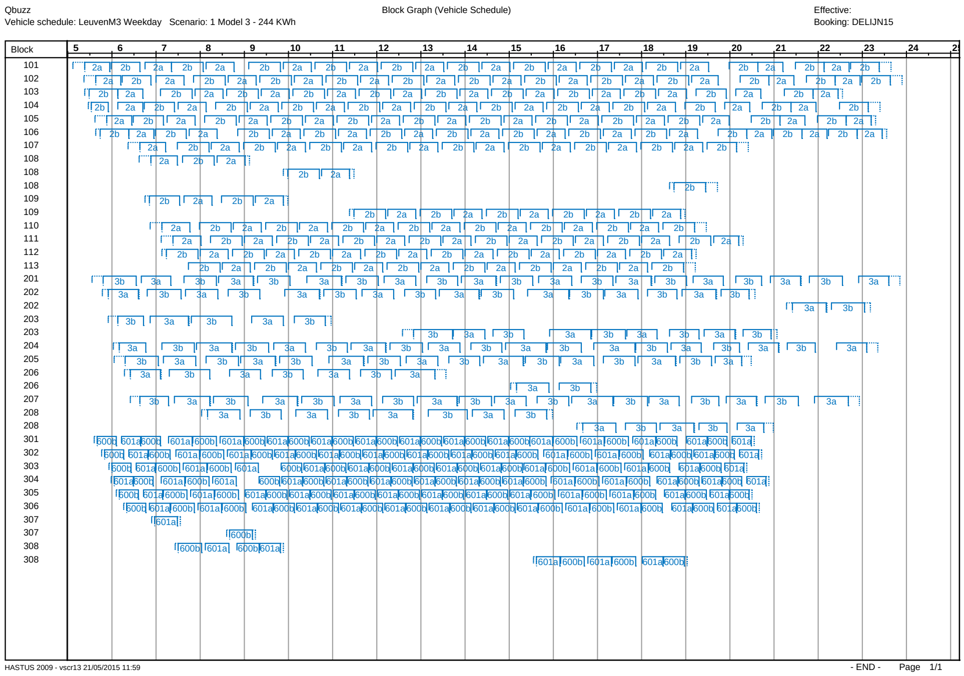

Figure 12: Vehicle schedule for the urban service of Leuven, using Model 3(Column Generation) for optimization and a battery capacity of 244 KWh. On the horizontal axis, the time of day is denoted. Every row shows the schedule of one vehicle. The numbers below the trips denote the route and direction of the trip.

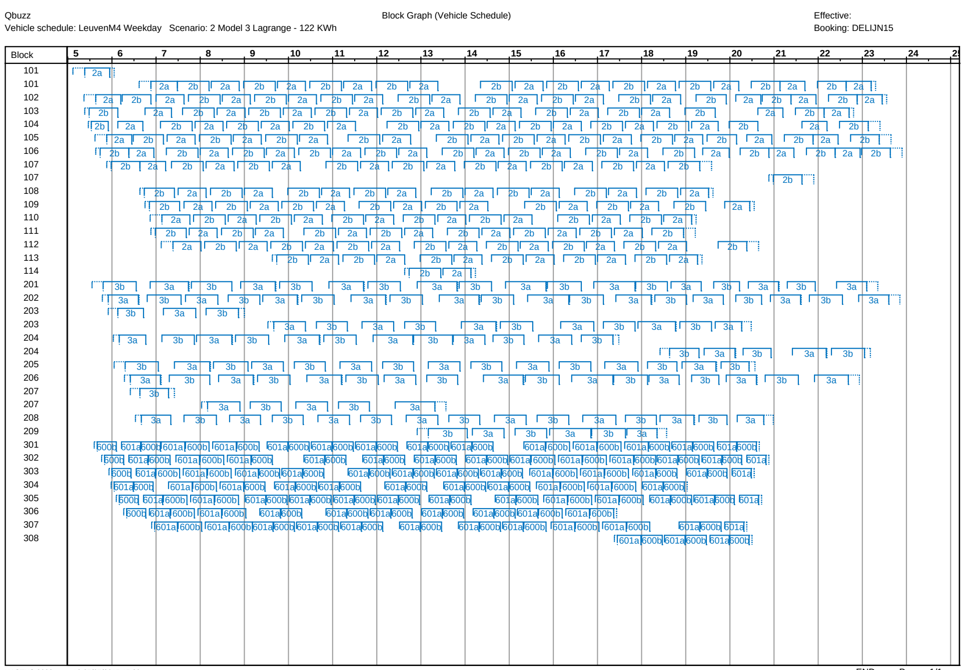

Figure 13: Vehicle schedule for the urban service of Leuven, using Model 3(Column Generation with Langrangean Relaxation) for optimization and a battery capacity of 122 KWh. On the horizontal axis, the time of day is denoted. Every row shows the schedule of one vehicle. The numbers below the trips denote the route and direction of the trip.

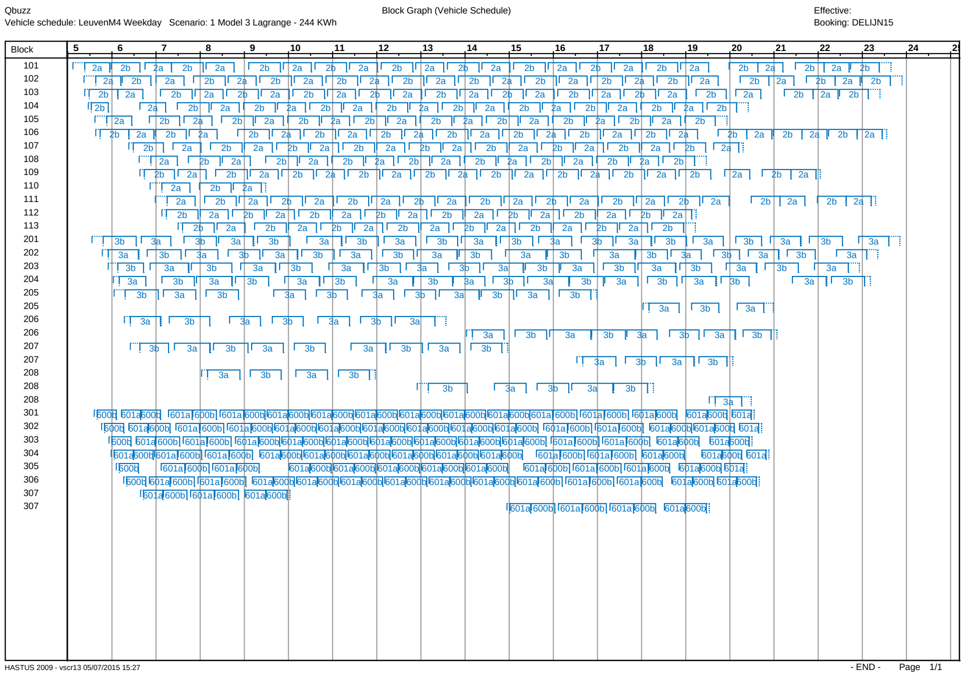

Figure 14: Vehicle schedule for the urban service of Leuven, using Model 3(Column Generation with Langrangean Relaxation) for optimization and a battery capacity of 244 KWh. On the horizontal axis, the time of day is denoted. Every row shows the schedule of one vehicle. The numbers below the trips denote the route and direction of the trip.

Comparing the traditional vehicles schedule with the one of model 1 and 122 KWh in Figure 7 shows that virtually every vehicle will run out of electricity if the 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 2, 3b and 3a. It was to be expected that model 2 gives worse results than model 1, because model 2 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.

5 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. We also see that in most cases, extra vehicles are needed. For this, we defined the e-VSP in Section 2

We have proposed three models to solve the e-VSP and tested them on four datasets. Comparing Model 1 to Model 2 shows us that both give us a solution with the same number of EVs, but that the solution of Model 2 needs more waiting hours than Model 1. This waiting time is for the majority used for charging the vehicles. Because Model 2 is an approximation of Model 1, it was to be expected that the results were less optimal than in Model 1.

Model 3a and 3b solve the problem faster than models 1 and 2, but the results are worse. However, this could be due to the small size of the instances. From models 3a and 3b, 3b outperforms 3a in all aspects.

For smaller instances with a linear charging scheme, we advice model 1. For non-linear charging schemes, model 2 performs best. Although model 3b gives generally more expensive results than models 1 and 2, this one may be chosen when large datasets are used or when speed is very important. Improving the quality of the results of model 3b can be subject of future research.

References

I. Buchmann. Battery university, 2014. URL `http://www.batteryuniversity.com`.

P Carraresi, L. Girardi, and M. Nonato. Network models, lagrangean relaxation and subgradients bundle approach in crew scheduling problems. *Computer-Aided Transit Scheduling, Proceedings of the Sixth International Workshop*, pages 188–212, 1995.

G. Desaulniers, J. Desrosiers, and M.M. Solomon. *Column Generation.* GERAD 25th Anniversary Series. Springer Science+Business Media, Inc., 2005. ISBN 978-0-378-25485-4.

J. Desrosiers, J.B. Gauthier, and M.E. Lübbecke. Row-reduced column generation for degenerate master problems. *European Journal of Operational Research*, 236:453–460, 2014. doi: 10.1016/j.ejor.2013.12.016.

R. Freling. *Models and Techniques for Integrating Vehicle and Crew Scheduling.* PhD thesis, Tinbergen Institute, Erasmus University Rotterdam, 1997.

D. Huisman, R. Jans, M. Peters, and A.P.M. Wagelmans. Combining column generation and lagrangean relaxation. In G. Desaulniers, J. Desrosiers, and M.M. Solomon, editors, *Column Generation*, GERAD 25th Anniversary Series, pages 247–270. Springer, New York, 2005.

B. Lawson. Electropaedia, battery and energy technologies, 2014. URL `http://www.mpoweruk.com`.

A-S. Pepin, G. Desaulniers, A. Hertz, and D. Huisman. A comparison of five heuristics for the multiple depot vehicle scheduling problem. *Journal of Scheduling*, 12:17–30, 2009.